ASBRU: A TASK-SPECIFIC, INTENTION-BASED, AND TIME-ORIENTED LANGUAGE FOR REPRESENTING SKELETAL PLANS

Silvia Miksch [1)]
Yuval Shahar [2)]
Peter Johnson [2)]

[1)] Vienna University of Technology
Institute of Software Technology
Resselgasse 3/E188, A-1040 Vienna, Austria, Europe
Email: silvia@ifs.tuwien.ac.at

[2)] Section on Medical Informatics, Medical School Office Building, x215
Stanford University, Stanford, CA 94 305 - 5479, USA
Email: {shahar, pdj}@camis.stanford.edu

in: Motta, E.; Harmelen, F. v.; Pierret-Golbreich, C.; Filby, I.; Wijngaards, N. (eds.), *7th Workshop on Knowledge Engineering: Methods & Languages (KEML-97)*, Milton Keynes, UK, 1997.

Abstract: Skeletal plans are a powerful way to reuse existing domain-specific procedural knowledge. They are instantiated and refined dynamically over time. In the **Asgaard** project, we are investigating 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. The underlying requirement to develop task-specific problem-solving methods is a *modeling language*. We developed a time-oriented machine-readable language, called **Asbru**, to represent and to annotate durative skeletal plans based on the task-specific ontology. During the design phase of plans, Asbru allows to express durative actions and plans caused by durative states of an observed agent. We represent explicitly the intentions underlying these plans as temporal patterns to be maintained, achieved or avoided. The result is an uniformly represented and organized plan-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.

3.1 INTRODUCTION: AUTOMATED SUPPORT OF SKELETAL PLAN DESIGN AND EXECUTION

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.

Execution of skeletal plans often involves an interpretation by one agent of plans that have been designed by another. We are interested in problems that occur while trying to provide several types of automated support to a human interpreting agent. Automated support includes tasks such as verification and validation of plans, assessment of the applicability of the plan to a particular state of the world, guidance in proper execution of that plan, monitoring of the execution process, assessment of the results of the plan, critiquing the execution process and its results, and assistance in the modification of the original plan [Shahar et al., 1996]. We are focusing on domains that are time-oriented with respect to both external states and plan actions, and that might require intermittent execution of plans over multiple (possibly disjoint) periods of time, an uncommon requirement in classical planning models. An urgent requirement to perform these tasks is an adequate language for representing durative skeletal plans.

We demonstrate our ideas in the context of the medical domain. However, as can readily be seen, the model presented is quite general. In the following we discuss the state-of-the-art of the influential components to our approach.

3.1.1 Clinical Guidelines and Protocols

Clinical guidelines are a set of schematic plans for management of patients who have a particular clinical condition (e.g., insulin-dependent diabetes). **Clinical protocols** are a more detailed version of clinical guidelines, often used when guidelines need to be applied uniformly to enable statistical analysis of the outcomes (e.g., experimental chemotherapy protocols for cancer therapy). The application of clinical guidelines and protocols by human care providers involves collecting and interpreting considerable amounts of data over time, applying standard treatment plans in an episodic fashion, and revising those plans when necessary. The guidelines often involve implicit assumptions about the knowledge of the agent executing the plans, both in the data-interpretation and in the treatment-planning phases. Thus, clinical guidelines can be viewed as a shared library of highly reusable skeletal reactive plans, whose details need to be refined and executed by a reactive planner over significant periods of time when applied to a particular patient [Tu et al., 1989] .

Clinical guidelines are often ambiguous or incomplete. For example, a diabetes guideline might recommend a therapy target without any specific recommendations on ways to achieve it, or might suggest the use of a drug without a precise dose. Physicians often do not adhere to protocols, believing their actions to be closer to the intentions of the protocol designers [Hickam et al., 1985]. An automated assistant should recognize cases in which the care-provider's actions adhere to the overall intentions, and continue offering useful advice even if the guideline is not followed literally.

3.1.2 Related Approaches: Automated Support

During the past 15 years, there have been several efforts to create automated reactive planners to support the process of protocol-based care over significant periods of time. In the *prescriptive* approach, active interpretation of the guidelines is given; examples include ONCOCIN [Tu et al., 1989] in the oncology domain, T-HELPER [Musen et al., 1992] in the AIDS domain, and DILEMMA [Herbert et al., 1995], EON [Musen et al., 1996], and the European PRESTIGE Health-Telematics project, as general architectures. In the *critiquing* approach, the program critiques the physician's plan rather than recommending a complete one of its own. This approach concentrates on the user's needs

and exploits the assumption that the user has considerable domain-specific knowledge [Miller, 1986]. A task-specific architecture implementing the critiquing process has been generalized in the HyperCritic system [Van der Lei and Musen, 1991]. Task-specific architectures assign well-defined problem-solving *roles* to domain knowledge and facilitate acquisition and maintenance of that knowledge.

Several approaches to the support of guideline-based care encode guidelines as elementary state-transition tables or as situation-action rules dependent on the electronic medical record [Sherman et al., 1995], but do not include an intuitive representation of the guideline's clinical logic, and have no semantics for the different types of clinical knowledge represented. Other approaches permit hypertext browsing of guidelines via the World Wide Web [Barnes and Barnett, 1995; Liem et al., 1995], but do not use the patient's electronic medical record.

None of the current guideline-based-care systems have a sharable representation of guidelines that (1) has knowledge roles specific to the guideline-based-care task, (2) is machine and human readable, and (3) allows data stored in an electronic patient record to invoke an application that directly executes the guideline's logic and related tasks, such as critiquing.

3.1.3 Related Approaches: Modeling Languages

On the one hand, workers in medicine and medical informatics have recognized the importance of protocol-based care to ensure a high quality of care since the 1970s. A group of investigators, working through the American Society for Testing and Materials (ASTM), has defined a standard procedural language, known as the Arden syntax [Hripcsak et al., 1994]. The Arden syntax 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. This new standard has significant limitations: The language currently supports only atomic data types, lacks a defined semantic for making temporal comparisons or for performing data abstraction, and provides no principled way to represent clinical guidelines that are more complex than individual situation-action rules [Musen et al., 1995]. Therefore the Arden syntax is not applicable for our purposes.

On the other hand, computer-oriented knowledge interchange languages (e.g., KIF [Genesereth and Fikes, 1992], CML [Bobrow et al., 1996]), ontologies or models for knowledge sharing (e.g., [Gruber, 1993; Guarina and Giaretta, 1995; Steve et al., 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 dynamic changing environments, like medical domains: (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) parallel and continuous execution of plans is necessary. The requirements of plan specifications in clinical domains [Tu et al., 1989; Uckun, 1996] are often a superset of the requirements in typical toy-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, we have developed a time-oriented, intention-based, and sharable language, called **Asbru**. The Asbru language is part of the **Asgaard**[*)] project [Shahar et al., 1996], in which we are developing task-specific problem-solving methods that perform design, execution, and critiquing tasks in medical domains.

In the following, we will introduce the syntax and the semantics of the Asbru language. We will explain Asbru using a medical example as illustration, namely a guideline for controlled observation and treatment of **noninsulin-dependent gestational diabetes mellitus (GDM type II)**. In the first appendix A we will list the complete Asbru syntax in Backus-Naur form (BNF). In the second appendix we will show the complete example - parts are used during the paper - in three version: verbal, graphical, and in Asbru.

3.2 THE ASBRU LANGUAGE: BASIC CONCEPTS

Asbru can be used to design specific plans as well as support the performance of different reasoning and executing tasks. During the design phase of plans, Asbru provides a powerful mechanism 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 of plan. 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.

3.2.1 Meaning of "Intention-based"

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]. We view intentions as temporally extended goals at various abstraction levels [Bacchus and Kabanza, 1996]. Intentions are *temporal patterns* of actions or states, to be maintained, achieved, or avoided.

For example, during therapy of a diabetes patient, hyperglycemia (a higher than normal level of blood glucose) is detected for the second time in the same week around bedtime. The diabetes-guideline's prescribed action might be to increase the dose of the insulin the patient typically injects before dinner. (Insulin reduces the level of blood glucose.) However, the health-care provider recommends reduction of the patient's carbohydrate intake (e.g., bread) during dinner. This action seems to contradict the prescribed action. Nevertheless, the underlying state intentions of both actions are the same: "avoid more than two episodes of hyperglycemia per week.". Increasing the dose of insulin decreases the value of the blood-glucose level directly (the prescribed action in the guideline), while decreasing the value of the same clinical parameter by reducing the magnitude of an action (i.e., ingestion of carbohydrates) that increases its value (the action of the health-care provider). Therefore, the provider is still following the intention of the protocol.

3.2.2 Meaning of "Temporal Pattern" and "Time annotations"

Intentions, world states, and prescribed actions are **temporal patterns**. A temporal pattern is

(a) a *parameter proposition*: a parameter (or its abstraction), its value, a context, and a time annotation (e.g., the *state* abstraction of the blood-glucose parameter is HIGH, as defined in the context of therapy for GDM type II, during a certain time period);

(b) a *combination* of multiple parameter propositions [Shahar and Musen, 1996]; or

(c) a *plan-state* associated to an instantiated plan (plan pointer) and a time annotation.

[*)] In Norse mythology, Asgaard was the home and citadel of the gods, corresponding to Mount Olympus in Greek mythology. It was located in the heavens and was accessible only over the rainbow bridge, called Asbru (or Bifrost).

The value can be a single value (or the correspondent level of abstraction-value), a value range, or the Boolean constraint "NOT" or "OR" to bound the value space. The context can be a single context (like GDM type II), a set of contexts, or a wildcard "*" for any context. The combination of multiple parameter propositions are specified either by Boolean, temporal, or value constraint or by constraint higher order, like counting the appearance of a particular temporal pattern.

The **time annotation** we use allows a representation of uncertainty in starting time, ending time, and duration [Dean et al., 1995; Rit, 1986]. The time annotation supports multiple time lines (e.g., different zero-time points and time units) by providing *reference annotations*. The reference annotation can be an absolute reference point, a reference point with uncertainty (defined by an uncertainty region), a function of a previously executed plan instance (e.g., start plan instance A1 20 minutes after having completed plan instance B1), or a domain-dependent time point variable (e.g., CONCEPTION). We define temporal shifts from the reference annotation 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) or domain-dependent units (e.g., GESTATIONAL-WEEKS). Thus, our temporal annotation is written as ([ESS, LSS], [EFS, LFS], [MinDu, MaxDu], REFERENCE). Figure 3.1 illustrates our time annotation. ESS, LSS, EFS, LFS, MinDu, and MaxDu can be "unknown" or "undefined" to allow incomplete time annotation. Also, in cases such as ([T1 HOURS, T1 HOURS], [T2 HOURS, T2 HOURS], [_, _], REFERENCE), the time interval has exact starting and finishing times, T1 and T2, respectively. Therefore, the duration should not be specified, because (duration = T2 - T1) and maximal duration is equal to minimal duration.

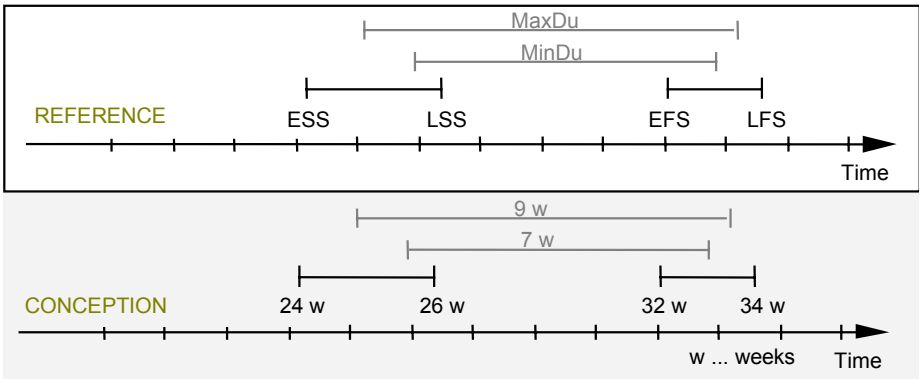

Figure 3.1 A schematic illustration of the Asbru time annotations. The upper part of the figure presents the generic annotation. The lower part shows a particular example representing the time annotation [[24 WEEKS, 26 WEEKS], [32 WEEKS, 34 WEEKS], [7 WEEKS, 9 WEEKS], CONCEPTION]), which means "starts 24 to 26 weeks after conception, ends 32 to 34 weeks after the conception, and lasts 7 to 9 weeks." REFERENCE = reference annotation, ESS = earliest starting shift, LSS = latest starting shift, EFS = earliest finishing shift, LFS = latest finishing shift, MinDu = minimal duration, MaxDu = maximal duration.

For example, the parameter proposition "high blood-glucose level of any type in the context of therapy for GDM type II for more than 7 days in the period from 24 conception weeks to delivery, using the estimated conception date as a reference point", is written in Asbru as:

```
(STATE(blood-glucose) HIGH GDM-Type-II [[24 WEEKS, 24 WEEKS],
       [DELIVERY, DELIVERY], [7 DAYS,_], CONCEPTION])
```

To allow temporal repetitions, we define sets of cyclical time points (e.g., MIDNIGHTS, which represents a set of midnights, where each midnight is exactly at 0:00 am, every 24 hours) and cyclical

time annotations (e.g., MORNINGS, which represents a set of mornings, where each morning starts at 8:00 am, ends at 11:00, and lasts at least 30 minutes). In addition, we allow short-cuts such as when a plan should start immediately at the current time, whatever that time is (using the symbol *NOW*), or when a condition should hold during the span of time over which the plan is executed, whatever that span is (using the symbol *).

For example,

```
MIDNIGHTS <- [0, 0 HOURS, 24 HOURS]
```

;; MIDNIGHTS is a set of cyclical time points

```
MORNINGS <-
   [[8 HOURS, 8 HOURS], [11 HOURS, 11 HOURS], [30 MINUTES, _], MIDNIGHTS]
```

;; MORNINGS is a set of cyclical time annotations or intervals with uncertainty concerning starting and ending that uses midnights as a reference annotation

All domain-dependent time annotations, units, and time abstractions have to be defined in advance to be applicable in all plans in the guideline-specification library. The definitions ensure that site-specific practice can be clarified and specified (e.g., DAYS start at 0:00 am or DAYS start at 7:00 am). We allow variable assignments of time units, domain-dependent time-points, time-intervals, and cyclical time abstractions.

In addition, a sampling-frequency argument specifies the frequency of sampling the external-world's data, such as when verifying the applicability of a particular plan. Thus, we define a sampling frequency for examining the plan's state-transition criteria (see Section 3.2.4).

Our notation enables the expression of interval-based intentions, states, and prescribed actions with uncertainty regarding starting, finishing, duration, and the use of absolute, relative, and even cyclical (with a predetermined granularity) reference annotations.

3.2.3 Decomposition and "Semantic" Stop-Condition

A (guideline) **plan** in the guideline-specification (plan) library is composed hierarchically, using the Asbru syntax, of a set of plans with arguments and time annotations. 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 calling of a program. The library also includes a set of **primitive** (nondecomposable) **plans** to perform interaction with the user or external devices such as asking the user for advice or retrieving particular information from the medical patient record (e.g., OBSERVE, GET-PARAMETER, ASK-PARAMETER, DISPLAY, WAIT)). Plans have return values.

3.2.4 Plan States and State-Transition Criteria

During the execution phase, an applicable plan is instantiated. A set of mutually exclusive **plan states** describes the actual status of the plan during execution. Particular **state-transition criteria** specify transition between neighboring plan states. Figure 3.2 illustrates the different plan states and their corresponding transition criteria mentioned on the arrows. The meaning of the state-transition criteria is explained in Section 3.3. We distinguish between plan states were an instantiation of a plan is needed (right-hand side of Figure 3.2) and between plan states without any instantiation of a plan during execution (left-hand side of Figure 3.2) . For example, if a plan has been activated, it can only be completed, suspended, or aborted depending on the corresponding criteria. The gray triangle on the right-hand side of Figure 3.2 includes the three basic states and associated transition criteria; these should always be defined. The `suspended` state is optional and is available for more complex plan types. The `restarted` state is defined implicitly: first, abort an activated plan and then restart it

from the pre-plan instantiation `considered`. The state-transition conditions are explained below. Generic library plans (i.e., plan types) also have states, such as `considered`, `possible`, `rejected`, and `ready`, that determine if a plan type is applicable and whether a plan instance can be created.

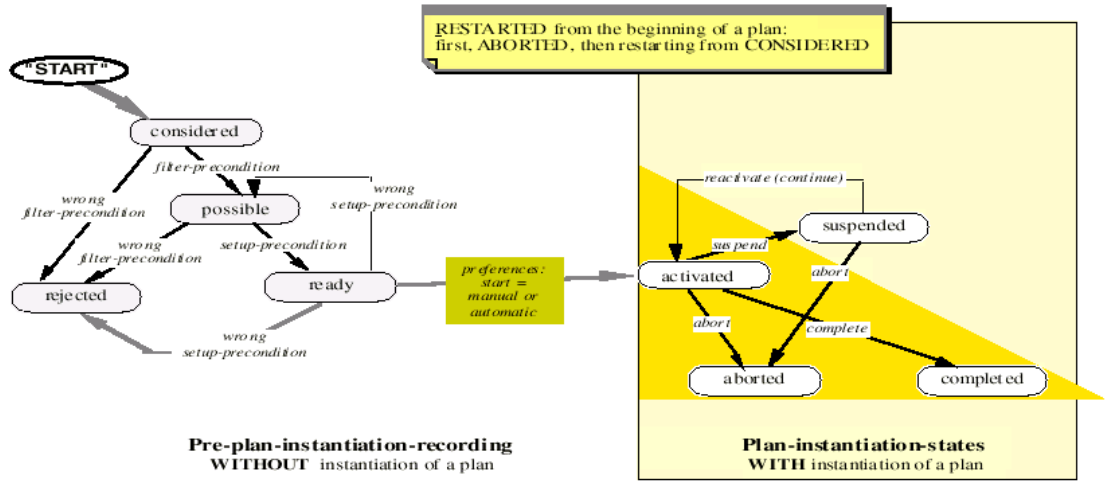

Figure 3.2 The various plan-instance states and associated state-transition criteria in Asbru.

3.3 COMPONENTS OF ASBRU

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** which describes the actions to be executed. The general arguments, the time annotation, and all components are optional. A subplan has the same structure (Figure 3.3).

For example, during the observation and the treatment period of GDM the pregnant woman gets instructions from the health care provider to execute several tasks in parallel. Each task takes place at a particular time point and has different frequency and duration (see Figure 3.4). In the current scenario, the observation period starts after GDM Type-II (noninsulin-dependent) was detected in third trimester pregnancy, as tested by a glucose tolerance test (GTT). In the remaining period of time until delivery, medical screening is executed in conjunction with medication management, nutrition management, observation of the mother's body weight gain, stress management, fetal evaluation, exercise level, and the need for insulin. A suitable component can be added in the case of insulin-dependent (Type-I) diabetes. The various broken lines in Figure 3.4 illustrate schematically the respective durations and frequencies of each task that should be performed until delivery. During this observation and treatment period the patient has to schedule meetings with the health care providers regularly and to present the self-acquired measurements and her reactions to particular health conditions. The health care providers may add new therapeutic tasks (e.g., injection of insulin) or adapt previous ones. Therefore, both the self-observed data and the applied clinical guidelines change according to the new instructions. (These adaptations are not shown in Figure 3.4.)

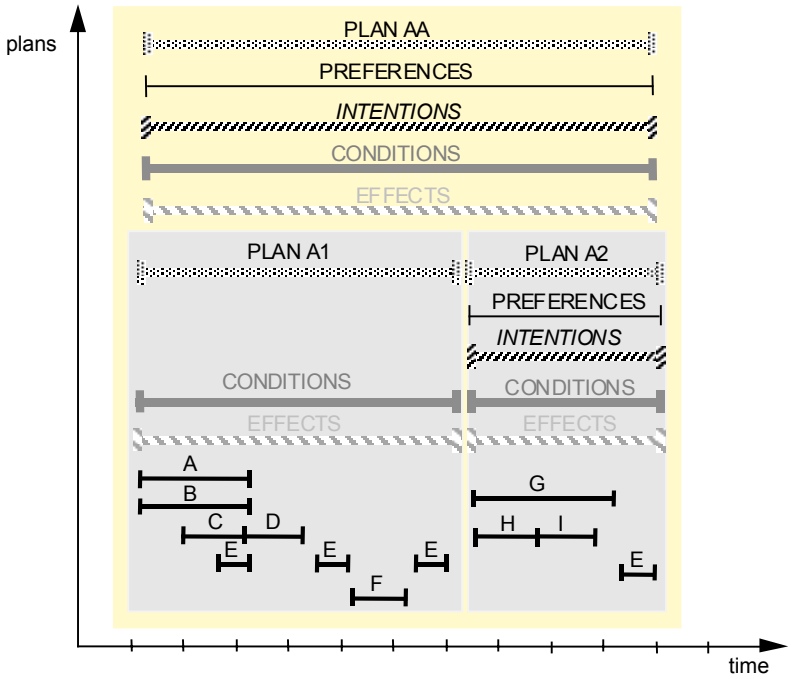

Figure 3.3 Graphical representation of a clinical-guideline specification represented in Asbru. Plan AA is of a sequential type and includes plans A1 and A2 in sequence; plan A1 is of a concurrent type and includes plans such as A, B, and C, and cyclical plan E.

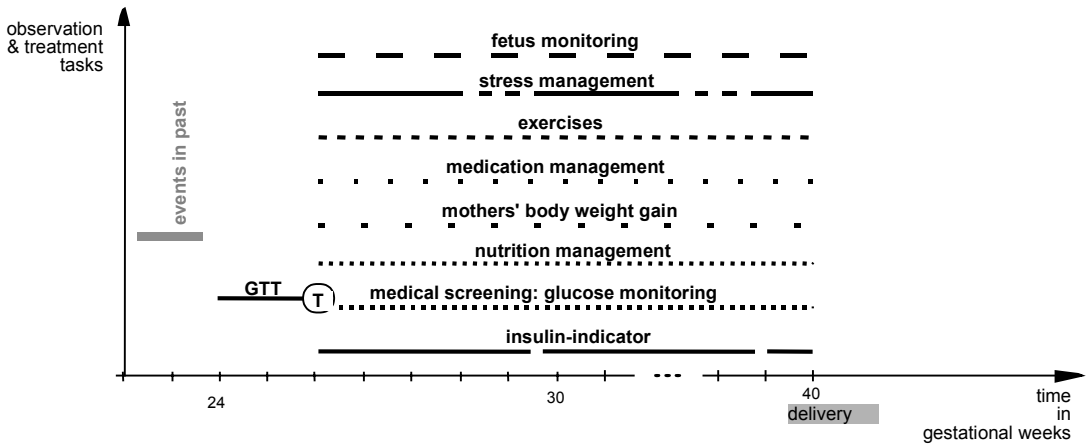

Figure 3.4 Observation and treatment tasks.

The relationship between the various shared task-specific knowledge roles and the Asbru language syntax is shown in Table 3.1.

Table 3.1 Relationship of Asbru syntactic elements to the task-specific knowledge roles.

Task-Specific Knowledge Roles	Syntactic Element in Asbru
Preferences: constrain the selection of a plan	`PREFERENCES (STRATEGY, UTILITY, LOOK-AHEAD, SELECT-METHOD, RESOURCES, START-CONDITION)`
Intended intermediate state: pattern that is intended to hold during plan execution	`INTENTION:INTERMEDIATE-STATE`
Intended overall states: pattern that is intended to hold at the end of plan execution	`INTENTION:OVERALL-STATE`
Intended intermediate actions: action pattern that is intended hold during plan execution	`INTENTION:INTERMEDIATE-ACTION`
Intended overall actions: action pattern that is intended to hold at the end of plan execution	`INTENTION:OVERALL-ACTION`
Filter preconditions that need to be true for the plan to be applicable	`FILTER-PRECONDITIONS`
Setup preconditions that need to be achieved so that the plan can start	`SETUP-PRECONDITIONS`
Suspension conditions that cause the plan to be suspended	`SUSPEND-CONDITIONS`
Restarting conditions that restart a suspended plan	`RESTART-CONDITION`
Abort conditions that abort the plan	`ABORT-CONDITIONS`
Completion conditions that determine when the plan is completed	`COMPLETE-CONDITIONS`
Prescribed actions	`PLAN-BODY`
Effects of plans in relation to measurable parameters	`ARGUMENT-DEPENDENCIES` `PLAN-EFFECTS`

We shall now examine in more detail each of the components represented in Asbru.

3.3.1 Preferences

Preferences bias or constrain the selection of a plan to achieve a given goal and express a kind of behavior of the plan. We distinguish between:

(1) **Strategy**: a general strategy for dealing with the problem (e.g., aggressive, normal);

(2) **Utility**: a set of utility measures (e.g., minimize the cost or inconvenience);

(3) **Select-method**: a matching heuristic for the applicability of the whole plan (e.g., exact-fit);

(4) **Resources**: a specification of prohibited or obligatory resources (e.g., in certain cases of treatment of a pulmonary infection, surgery is prohibited and antibiotics must be used);

(5) **Start-conditions:** an indication whether transition from a `ready` generic plan to the `started` state of an actual plan instance is automatic (after applying the *filter* and *setup* preconditions—see below) or requires approval of the user.

For example, the match in the filter conditions needs to be exact and the plan starts as soon as it is in a ready state, is represented as

```
(PREFERENCES
     (SELECT-METHOD EXACT-FIT)
     (START-CONDITION AUTOMATIC))
```

3.3.2 *Intentions*

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. We define four categories of intentions:

(1) **Intermediate state:** the state(s) that should be maintained, achieved, or avoided during the applicability of the plan (e.g., weight gain levels are slightly low to slightly high);

(2) **Intermediate action:** the action(s) that should take place during the execution of the plan (e.g., monitor blood glucose once a day);

(3) **Overall state pattern:** the overall pattern of states that should hold after finishing the plan (e.g., patient had less than one high glucose value per week);

(4) **Overall action pattern:** the overall pattern of actions that should hold after finishing the plan (e.g., patient had visited dietitian regularly for at least three months).

For example,

```
(INTENTION:INTERMEDIATE-STATE
  (MAINTAIN STATE(mothers-body-weight-gain)
    (OR SLIGHTLY-LOW NORMAL SLIGHTLY-HIGH) GDM-Type-II
    [[24 WEEKS, 24 WEEKS], [DELIVERY, DELIVERY], [_,_], CONCEPTION]))


(INTENTION:OVERALL-ACTION
  (MAINTAIN visit-dietitian regularly GDM-Type-II
    [[24 WEEKS, 24 WEEKS], [DELIVERY, DELIVERY], [3 MONTHS,_],CONCEPTION])
```

3.3.3 *Conditions*

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. We do not directly determine conditions that should hold during execution. We specify different conditions that enable transition from one plan state into another (see Figure 3.2). 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. All conditions are optional.

We distinguish between:

(1) **Filter-preconditions**, which need to hold initially if the plan is applicable, but can not be achieved (e.g., patient is a female), and are necessary for a `possible` state;

(2) **Setup-preconditions**, which need to be achieved to enable a plan to start (e.g., patent had a glucose-tolerance test) and allow a transition from a `possible` plan to a `ready` plan;

(3) **Suspend-conditions**, which determine when a started plan has to be suspended (e.g., blood glucose has been high for at least four days); these implicitly what the planning literature calls *protection intervals* [Kambhampati et al., 1995], in which certain conditions need to hold;

(4) **Abort-conditions**, which determine when a started, suspended, or restarted plan has to be aborted (e.g., there is an insulin-indicator condition: the patient cannot be controlled by diet);

(5) **Complete-conditions**, which determine when a started or restarted plan has to be completed successfully (e.g., delivery has been performed);

(6) **Restart-conditions**, which determine when a suspended plan has to be restarted (e.g., blood glucose level is back to normal or is only slightly high); these can be seen as "automatic" or "internal" reactivation conditions: others might be imposed by the higher-level plan. The `restarted` state is defined implicitly: first, abort an activated plan and then restart it from the pre-plan instantiation `considered`.

For example,

```
(SUSPEND-CONDITIONS
   (STATE(blood-glucose) HIGH GDM-Type-II
      [[24 WEEKS,24 WEEKS], [DELIVERY, DELIVERY], [4 DAYS,_], CONCEPTION]
   ;; suspend if the glucose is HIGH during this period
   (SAMPLING-FREQUENCY 24 HOURS))))

(ABORT-CONDITIONS (OR ACTIVATED SUSPENDED)
   (insulin-indicator-conditions TRUE GDM-Type-II *
   (SAMPLING-FREQUENCY 24 HOURS)))
   ;; abort simpler plan if there is an indication for need of insulin
```

3.3.4 *Effects*

Effects describe the functional relationship between the plan arguments and measurable parameters (e.g., the *dose* of insulin is inversely related to the level of blood glucose) or the overall effect of a plan on parameters (e.g., administration of insulin decreases the blood glucose). Effects have a likelihood annotation—a probability of occurrence.

For example, in the context of GDM, the dose argument of the insulin-administration plan has a negative-monotonic relationship to the blood-glucose level, for any reaction time. This effect relation (ignoring in this case the temporal span and likelihood) is written as

```
(ARG-DEPENDENCY (dose GDM glucose_level NEGATIVE-MON
              [[_, _], [_, _], [[_, _], *NOW*], _ ))
```

The overall effect of the plan in the context of GDM decreases the blood-glucose level, the reaction time is between 10 and 60 minutes, and the likelihood is 0.97. This overall effect is written as

```
(PLAN-EFFECTS (GDM glucose_level DEC
       [[_, _], [_, _], [10 MINUTES, 60 MINUTES], *NOW*] 0.97))
```

3.3.5 *Plan-Body*

The plan body is a set of plans to be executed in parallel, in sequence, in any order, or in some frequency. We distinguish among several types of plans: *sequential*, *concurrent*, and *cyclical*. Only one type of plan is allowed in a single plan body. 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. We distinguish two dimensions for classification of sequential or (potentially) concurrent plans: the number of plans that should be completed to enable continuation and the order of plan execution. Table 3.2 summarizes the dimensions of the two plan types. Using the two dimensions, we define the operators DO-ALL-TOGETHER, DO-SOME-TOGETHER, DO-ALL-ANY-ORDER, DO-SOME-ANY-ORDER, DO-ALL-SEQUENTIALLY. The continuation condition specifies the names of the plans that must be completed to proceed with the next steps in the plan. For example,

```
(DO-ALL-TOGETHER
  ;; a sequential plan type in which continuation depends on completion of
  the preceding plan
     (glucose-monitoring)
     (nutrition-management)
     (observe-insulin-indicators))
     ;; three subplans; the plan body of each can be of any type
```

A cyclical plan (an EVERY clause) includes a plan that can be repeated, and optional temporal and continuation arguments that can specify its behavior. *Start* and *end* specify a starting and ending time point. *Time base* determines the time interval over which the plan is repeated and the start time, end time, and duration of the particular plan instance in each cycle (e.g., starting with the first Monday's morning, until next Tuesday's morning, perform plan A every morning for 10 minutes). The *times-*

completed argument specifies how many times the plan has to be completed to succeed and the *times-attempted* argument specifies how many attempts are allowed. Obviously, the number of attempts must be greater or equal to the number of completions. A temporal pattern can be used as a stop condition of the cyclical plan. Finally, the plan itself is associated with its own particular arguments (e.g., dose). The start time, the time base, and the plan name are mandatory to the specification of a cyclical plan; the other arguments are optional.

For example, consider the following plan: "Administer 5 units of insulin every morning starting with the first morning following the initiation of the plan." This plan could be written as:

```
(EVERY
     (START (FIRST(MIDNIGHT) after (ACTIVATED *self*))
     (TIME-BASE [[8 HOURS, 8 HOURS], [11 HOURS, 11 HOURS], [_,_],
        MIDNIGHTS]
     administer-insulin 5)
END-EVERY )
```

;; Note that *morning* is a cyclical time annotation and is represented in this case as the interval 8 a.m. to 11 a.m., expressed as a time shift from the cyclical set of time points MIDNIGHTS. The duration of administration of insulin in this case is not constrained, but it could be. Note also that no stop condition is defined in this case; the plan would continue indefinitely.

Table 3.2 Categorization of plan types according to continuation conditions and ordering constraints.

Continuation Condition --> Ordering Constraints	All plans should be completed to continue	Some plans should be completed to continue
Start together	DO-ALL-TOGETHER (no continuation-condition; all plans must complete)	DO-SOME-TOGETHER (continuation-conditions specified as subset of plans)
Execute in any order	DO-ALL-ANY-ORDER (no continuation-condition; all plans must complete)	DO-SOME-ANY-ORDER (continuation-conditions specified as subset of plans)
Execute in total order (sequence)	DO-ALL-SEQUENTIALLY (no continuation-condition; all plans must complete)	------

3.4 CONCLUSION

We investigated within the Asgaard project in 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. An adequate language to represent time-oriented skeletal plans is imperative to proceed with our kind of automated support. Existing approaches are unable to handle the characteristics of dynamically changing environments, such as durative events and actions, uncertainty and variability in the utility of available actions, concurrent and cyclical plans, and actions which do not occur instantaneously.

We have suggested a modeling language, called Asbru, which places a particular emphasis on an expressive representation for time-oriented actions and world states. Our 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. It is based on different granularities and reference points to support multiple time lines. It has a rich set of parallel, sequential, and iterative operators and effects, which enable to express complex procedures. We represent explicitly the intentions underlying skeletal plans as temporal patterns to be maintained, achieved or avoided. Clear

semantics for the task-specific knowledge facilitate acquisition and maintenance. The result is an uniformly represented and organized plan-specification library.

Representing complex skeletal plans, such as clinical guidelines, and the intentions underlying them in a standard, sharable, acquirable, machine-readable, and machine-interpretable form is imperative for sharing execution plans and for useful, flexible automated assistance in the execution of these plans.

In the Asgaard project, we are currently focusing on the development of the execution, plan recognition, and critiquing problem-solving methods.

Acknowledgmentss

This work has been supported by grants LM05708 and LM06245 from the National Library of Medicine, IRI-9528444 from the National Science Foundation, and DARPA Grant N66001-94-D-6055. Computing resources were provided by the Stanford CAMIS project, funded under Grant No. LM05305 from the National Library of Medicine. Silvia Miksch was supported by "Erwin Schrödinger Auslandstipendium, Fonds zur Förderung der wissenschaftliche Forschung", J01042-MAT. The authors thank Anne Regenstein, Gretchen D. Flanagan, and Lylia Needham (California diabetes and pregnancy program "Sweet Success") for their fruitful cooperation and medical contributions.

References

[Bacchus and Kabanza, 1996] Bacchus, F. and Kabanza, F. (1996). Planning for Temporally Extended Goals. Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96), pages 1215–1222, AAAI Press/The MIT Press, Menlo Park, CA.

[Barnes and Barnett, 1995] Barnes, M. and Barnett, G. O. (1995). An Architecture for a Distributed Guideline Server. In Gardner, R. M., editor, Proceedings of the Annual Symposium on Computer Applications in Medical Care (SCAMC-95), pages 233-237, Hanley & Belfus.

[Bobrow et al., 1996] Bobrow, D., Falkenhainer, B., Farquhar, A., Fikes, R., Forbus, K., Gruber, T., Iwasaki, Y., and Kuipers, B. (1996). A Compositional Modeling Language. In Iwasaki, Y., and Farquhar, A., editors, Proceedings of the Tenth International Workshop for Qualitative Reasoning (QR-96), pages 12-21, AAAI Press, AAAI Technical Report WS-96-01, Melno Park.

[Bratman, 1987] Bratman, M. E. (1987). Intention, Plans and Practical Reason. Harvard University Press, Cambridge, MA.

[CDAPP, 1992] CDAPP (1992). "Sweet Success" California Diabetes and Pregnancy Program, Guidelines for Care.

[Dean et al., 1995] Dean, T., Kaelbling, L. P., Kirman, J., and Nicholson, A. (1995). Planning under Time Constraints in Stochastic Domains. Artificial Intelligence, Special Volume on Planning and Scheduling, 76(1-2):35-74.

[Friedland and Iwasaki, 1985] Friedland, P. E. and Iwasaki, Y. (1985). The Concept and Implementaion of Skeletal Plans. Journal of Automated Reasoning, 1(2):161-208.

[Genesereth and Fikes, 1992] Genesereth, M. R. and Fikes, R. E. (1992). Knowledge Interchange Format, Version 3.0 Reference Manual. Computer Science Department, Stanford University, Technical Report Logic-92-1.

[Gruber, 1993] Gruber, T. R. (1993). Toward Principles for the Design of Ontologies Used for Knowledge Sharing, Knowledge Systems Laboratory. Computer Science Dept., Stanford University, Stanford, CA 94305, USA, TR KSL 93-04.

[Guarina and Giaretta, 1995] Guarina, N. and Giaretta, P. (1995). Ontologies and Knowledge Bases. In Mars, N. J. I., editor, Towards Very Large Knowledge Base, Amsterdam: IOS Press.

[Herbert et al., 1995] Herbert, S. I., Gordon, C. J., Jackson-Smale, A., and Renaud Salis, J.-L. (1995). Protocols for Clinical Care. Computer Methods and Programs in Biomedicine, 48:21-26.

[Hickam et al., 1985] Hickam, D. H., Shortliffe, E. H., Bischoff, M. B., Scott, A. C., and Jacobs, C. D. (1985). A Study of the Treatment Advice of a Computer-Based Cancer Chemotherapy Protocol Advisor. Ann Intern Med., 103:928-36.

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

[Kambhampati et al., 1995] Kambhampati, S., Knoblock, C. A., and Yang, Q. (1995). Planning as Refinement Search: A Unified Framework for Evaluating Design Tradeoffs in Partial Order Planning. Artificial Intelligence, Special Volume on Planning and Scheduling, 76(1-2):167-238.

[Levinson, 1995] Levinson, R. (1995). A General Programming Language for Unified Planning and Control. Artificial Intelligence, Special Volume on Planning and Scheduling, 76(1-2):319-75.

[Liem et al., 1995] Liem, E. B., Obeid, J. S., Shareck, P. E., Sato, L., and Greenes, R. A. (1995). Representation of Clinical Practice Guidelines Through an Interactive World-Wide-Wb Interface. In Gardner, R. M., editor, Proceedings of the Annual Symposium on Computer Applications in Medical Care (SCAMC-95), pages 223-227, Hanley & Belfus.

[Miller, 1986] Miller, P. L. (1986). Expert Critiquing System: Practice-Based Medical Consultation by Computer. Springer, New York, NY.

[Musen et al., 1992] Musen, M. A., Carlson, C. W., Fagan, L. M., Deresinski, S. C., and Shortliffe, E. H. (1992). T-HELPER: Automated Support for Community-Based Clinical Research. In Frisse, M. E., editor, Proceedings of the Sixteenth Annual Symposium on Computer Applications in Medical Care (SCAMC-92), pages 719-23, McGraw Hill, New York, NY.

[Musen et al., 1995] Musen, M. A., Tu, S. W., Das, A. K., and Shahar, Y. (1995). A Component-Based Architecture for Automation of Protocol-Directed Therapy. In Barahona, P., Stefanelli, M., and Wyatt, J., editors, Proceedings of the Artificial Intelligence in Medicine Europe, 5th European Conference on Artificial Intelligence in Medicine Europe (AIME-95), pages 3-16, Springer, Berlin.

[Musen et al., 1996] Musen, M. A., Tu, S. W., Das, A. K., and Shahar, Y. (1996). EON: A Component-Based Approach to Automation of Protocol-Directed Therapy. Journal of the American Medical Information Association, 3(6):367-88.

[Pollack, 1992] Pollack, M. (1992). The Use of Plans. Artificial Intelligence, 57(1):43-68.

[Rit, 1986] Rit, J.-F. (1986). Propagating Temporal Constraints for Scheduling. Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86), pages 383-388, Morgan Kaufmann, Los Altos, CA.

[Shahar et al., 1996] Shahar, Y., Miksch, S., and Johnson, P. (1996). A Task-Specific Ontology for Design and Execution of Time-Oriented Skeletal Plans. In Gaines, B., editor, Proceedings of the Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop (Track KA for Temporal Reasoning and Planning).

[Shahar and Musen, 1996] Shahar, Y. and Musen, M. A. (1996). Knowledge-Based Temporal Abstraction in Clinical Domains. Artificial Intelligence in Medicine, Special Issue Temporal Reasoning in Medicine, 8(3):267-98.

[Sherman et al., 1995] Sherman, E. H., Hripcsak, G., Starren, J., Jender, R. A., and Clayton, P. (1995). Using Intermediate States to Improve the Ability of the Arden Syntax to Implement Care Plans and Reuse Knowledge. In Gardner, R. M., editor, Proceedings of the Annual Symposium on Computer Applications in Medical Care (SCAMC-95), pages 238-242, Hanley & Belfus.

[Steve et al., 1995] Steve, G., Gangemi, A., and Rossi-Mori, A. (1995). Modeling a Sharable Medical Concept System: Ontological Foundation in GALEN. In Barahona, P., Stefanelli, M., and Wyatt, J., editors, Proceedings of the Artificial Intelligence in Medicine, 5th European Conference on Artificial Intelligence in Medicine Europe (AIME-95), pages 411-12, Springer, Berlin.

[Tate et al., 1994] Tate, A., Drabble, B., and Kibry, R. (1994). O-Plan2: An Open Architecture for Command, Planning, and Controll. In Zweben, M., and Fox, M. S., editors, Intelligent Scheduling, Vol. 1, pp. 213-39, San Francisco, CA: Morgan Kaufmann.

[Tu et al., 1989] Tu, S. W., Kahn, M. G., Musen, M. A., Ferguson, J. C., Shortliffe, E. H., and Fagan, L. M. (1989). Episodic Skeletal-Plan Refinement on Temporal Data. Communications of the ACM, 32:1439-1455.

[Uckun, 1996] Uckun, S. (1996). Instantiating and Monitoring Skeletal Treatment Plans. Methods of Information in Medicine, 35:324-333.

[Van der Lei and Musen, 1991] Van der Lei, J. and Musen, M. A. (1991). A Model for Critiquing based on Automated Medical Records. Computers and Biomedical Research, 24(4):344-78.

APPENDIX A: ASBRU: BACKUS-NAUR FORM (BNF) SYNTAX

Notation	Meaning	Notation	Meaning
::=	is defined as	(expr)	grouping expr in syntax
|	or (disjunction)	[expr]	optional expr
blank	and (conjunction)	[t1, t2]	time/duration interval
< >	non-terminal symbol	+	one or more
A...Z, a...b	terminal symbol, <symbol>	*	zero or more

A.1: TIME ANNOTATION

```
<time-annotation>::= [<time-range>, <reference>] | *
                                          ;; short-cuts: * means any time annotation
<time-range> ::=
   [<time-measure>, <time-measure>], [<time-measure>, <time-measure>],
   [<time-measure>, <time-measure>]
<time-measure> ::= <number> <unit> |
                     <variable-domain-dependent-time-shift>
<unit> ::= SECONDS | MINUTES | HOURS | DAYS | WEEKS | MONTHS | YEARS |
       <variable-domain-dependent-time-unit>
<reference> ::= <reference-point> |<variable-domain-dependent-time-point> |
        (ACTIVATED <plan-pointer> | *self*) |
        (RESTARTED <plan-pointer> | *self*) |
        (SUSPENDED <plan-pointer>| *self*) |
        (ABORTED <plan-pointer> | *self*) |
        (COMPLETED <plan-pointer> | *self*)
<reference-point> ::=  <absolute-time-point> |
             (<absolute-time-point> ± <ε-region>)
<ε-region> ::= <number>
<plan-pointer> ::= "pointer to an instantiation of a plan (plan name)
   during execution time"
<absolute-time-point> ::=
   <date> | <domain-independent-time-point> | YY-MM-DD - HH:MM:SS |
   *NOW* | <variable-domain-dependent-time-point>
<set-of-cyclical-time-annotations> ::=
        [<time-range>, <set-of-cyclical-time-points>]
<set-of-cyclical-time-points> ::=
        [<absolute-time-point> <off-set> <frequency>]
<off-set> ::= <time-measure>
<frequency> ::= <time-measure>
<date> ::= <symbol> | YY-MM-DD - HH:MM:SS
<domain-independent-time-point> ::= <symbol> | *NOW*
<variable-domain-dependent-time-unit> ::= <symbol>
                                          ;; e.g., GESTATIONAL-WEEK
<variable-domain-dependent-time-point> ::= <symbol>   ;; e.g., DATE-OF-BIRTH
<variable-domain-dependent-time-interval> ::= <symbol> ;; e.g., LUNCH
```

A.2: PLAN DEFINITIONS

```
<plan-definition> ::=
   (  PLAN <plan-name> [(ARG <argument>+)]
        [<set-of-time-assignments>]
        [<set-of-preferences>]
        [<set-of-intentions>]
        [<set-of-conditions>]
        [<set-of-effects>]
        [<plan-body>]
        [<return-expression>]
   )
<plan-name> ::= <symbol>
<symbol> ::= A| ..|Z | a|..|b | ...
<argument> ::= <symbol>
<file-name> ::= <symbol>
```

```
<set-of-time-assignments> ::=
   INCLUDE <file-name> |                                   ;; global definition
   [(TIME-UNIT-ASSIGNMENT
      (<unit> <- <unit-definition>)+)]
   [(DOMAIN-DEPENDENT-TIME-ASSIGNMENT
        [(SHIFTS
           (<variable-domain-dependent-time-shift> <- <time-measure>)+)]
        [(POINT
           (<variable-domain-dependent-time-point>
                              <- <absolute-time-point>)+)]
        [(INTERVAL
           (<variable-domain-dependent-time-interval>
                 <- (<time-range>   <absolute-time-point>|
                      <variable-domain-dependent-time-point>))+)])]
   [(ABSTRACTION-ASSIGNMENT
      (CYCLICAL
        [(<variable-cyclical-time-points>
                 <- <set-of-cyclical-time-points>)+]
        [(<variable-cyclical-time-annotations>
                 <- <set-of-cyclical-time-annotations)+])]
```

```
<set-of-preferences> ::=
   (PREFERENCES  [(STRATEGIES <strategy>)]
                 [(UTILITIES <utility-measure>)]
                 [(LOOK-AHEAD <look-ahead-flag>)]
                 [(SELECT-METHOD <select-criteria>)]
                 [(RESOURCES <resource-constraint>+)]
                 [(START-CONDITION <started>)]
<strategy> ::= <symbol> ;; e.g., AGGRESSIVE, NORMAL
<utility-measure> ::= "defined by user - domain and cite specific"
<look-ahead-flag> ::=  <checknumber> | ALL
<checknumber> ::= 2| 3| ...
<select-criteria> ::= <symbol>|EXACT-FIT|ROUGHLY-FIT|ASK|DEFAULT-VALUE
```

```
<resource-constraint> ::=     [(PROHIBITED <set-of-resources>)]
                              [(RECOMMENDED <set-of-resources>)]
                              [(DISCORAGED <set-of-resources>)]
                              [(OBLIGATORY <set-of-resources>)]
                              [(RESPONSIBLE-ACTOR <role>+)]
<set-of-resources>) ::= (<resource> <time-annotation>)+
<resource> ::= <symbol>
<started> ::= MANUAL | AUTOMATIC
<role> ::= <symbol> ;; e.g., DOCTOR, NURSE, PATIENT
```

```
<set-of-intentions> ::=
  [(INTENTION:INTERMEDIATE-STATE <intention>+)] |
  [(INTENTION:INTERMEDIATE-ACTION <intention>+)] |

  [(INTENTION:OVERALL-STATE <intention>+)] |
  [(INTENTION:OVERALL-ACTION <intention>+)]
<intention> ::=
             (<intention-verb> <temporal-pattern> [<importance-measure>])
<intention-verb> ::= AVOID | MAINTAIN | ACHIEVE
<importance-measure> ::= 0 .. 1   ;; between 0 and 1
<temporal-pattern> ::=
     <parameter-proposition> |
     <pattern> |
     (PLAN-STATE <plan-pointer>
           ([<boolean-constraint>] <plan-state-value>+) <time-annotation>)
<parameter-proposition> ::=
  (<parameter> <value-description> <context> <time-annotation>)
<parameter> ::=
  <parameter-name> | <abstraction-function>(<parameter-name>)
<parameter-name> ::= <symbol>
<abstraction-function>(<parameter-name>) ::=
       STATE(<parameter-name>) |
       GRAD(<parameter-name>) |
       RATE(<parameter-name>)
<value-description> ::=
  <single-value> | <value-range> | (<boolean-constraint> <single-value>+ )
<single-value> ::= <state-value> | <grad-value> | <rate-value> | <number>
<value-range> ::= (<single-value>, <single-value>)
<state-value> ::= <number> | <symbol> | HIGH | NORMAL | LOW
<grad-value> ::= <symbol> | INC | DEC | SAME | NON-INC | NON-DEC |
  NON-MON |
<rate-value> ::= <number> <unit> | <symbol> | FAST | NORMAL |SLOW
<pattern> ::= <constraints> <parameter-proposition>+
<plan-state-value> ::= STARTED | SUSPENDED | RESTARTED | ABORTED |
  COMPLETED
<constraints> ::=   COUNT-APPEARANCE <number> |
                 ORDINAL-NUMBER <number> |
                 <boolean-constraint> |
                 <temporal-constraint> |
                 <value-constraint>
<boolean-constraint> ::= OR | NOT
<temporal-constraint> ::= <symbol< | BEFORE | AFTER
```

```
<value-constraint> ::= < | • | > | • | = | =/=
<context> ::= * | <symbol> | <set-of-contexts>  ;; short-cuts: * means any context
<set-of-contexts> ::= (<context>+)
```

```
<set-of-conditions> ::=
   [(FILTER-PRECONDITIONS <temporal-pattern>+)]
   [(SETUP-PRECONDITIONS <temporal-pattern>+)]
   [(ACTIVATED-CONDITIONS <activated-from-state>)]
   [(SUSPEND-CONDITIONS <temporal-pattern>+ <sampling-frequency> )*]
   [(ABORT-CONDITIONS <aborted-from-state> <temporal-pattern>+
                        <sampling-frequency> )*]
   [(COMPLETE-CONDITIONS <temporal-pattern>+ <sampling-frequency> )*]
   [(RESTART-CONDITIONS <temporal-pattern>+
                        <sampling-frequency> )]
<activated-from-state> ::= MANUAL | AUTOMATIC | SUSPENDED |
              (<boolean-constraint> <activated-from-state>+)
<aborted-from-state> ::= ACTIVATED | SUSPENDED |
              (<boolean-constraint> <aborted-from-state>+)
<sampling-frequency> ::=
   (SAMPLING-FREQUENCY <number> <unit>    | <variable-cyclical-time-points>
                       | <variable-cyclical-time-annotations>)
```

```
<set-of-effects> ::=
   [(ARG-DEPENDENCY (<argument> <context> <parameter>
        <relationship-function>  [<time-annotation>] <likelihood>)+]
   [(PLAN-EFFECTS (<context> <parameter>
        <direction-of-change>  [<time-annotation>] <likelihood>)+]
<likelihood> ::= "desirable probability"
<relationship-function> ::= <symbol> | POSITIVE-MON | NEGATIVE-MON
<direction-of-change> ::= <symbol> | INC | DEC | NORMAL
```

```
<plan-body> ::=
   <sequential-plan> |
   <parallel-plan> |
   <any-order-plan> |
   <cyclical-plan>
<sequential-plan> ::=
   (DO-ALL-SEQUENTIALLY <assign-statement>+ |
                        <plan-activation>+|
                        <variable-plan-schema>|
                        <variable-plan-schema-list>) |
                        <if-then-else>
<parallel-plan> ::=
   (DO-ALL-TOGETHER <plan-activation>+ |
                  <variable-plan-schema> |
                  <variable-plan-schema-list>) |
   (DO-SOME-TOGETHER  <plan-activation>+ |
                  <variable-plan-schema> |
                  <variable-plan-schema-list>
                  [CONTINUATION-CONDITION <plan-name>+])
                  <if-then-else>
```

```
<any-order-plan>::=
   (DO-ALL-ANY-ORDER  <plan-activation>+ |
                 <variable-plan-schema>
                 <variable-plan-schema-list>) |
   (DO-SOME-ANY-ORDER <plan-activation>+ |
                 <variable-plan-schema> |
                 <variable-plan-schema-list>
                 [CONTINUATION-CONDITION <plan-name>+]) |
                 <if-then-else>
<cyclical-plan> ::=
   (EVERY  (START <start-time>) [(END <end-time>)]
           (TIME-BASE [<time-range> <set-of-cyclical-time-points>] |
                    <variable-cyclical-time-annotations>)
           [(TIMES-COMPLETED <number-times-completed>)]
           [(TIMES-ATTEMPTS <max-number-of-attempts>
                 [(<min-time-attempt>, <max-time-attempt>)])]
           [(UNTIL-COND <temporal-pattern>)]
           <plan-name> [<plan-argument-value>+]
   END-EVERY )
<assign-statement> ::=
   (ASSIGN <variable-name> <plan-schema>) |
   (ASSIGN <variable-plan-schema-list>
           (SELECT-SOME <plan-schema>+
           [(SELECT-METHOD <select-criteria>)])) |
   (ASSIGN <variable-plan-schema>
           (SELECT-ONE <plan-schema>+
           [(SELECT-METHOD <select-criteria>)]))

<if-then-else> ::= (IF <temporal-pattern>      THEN <plan-activation>
                                          ELSE <plan-activation> )
<plan-activation> ::=
   (<plan-schema>   (ON-SUSPEND (<plan-activation> | <if-then-else>))
                 (ON-ABORT (<plan-activation> | <if-then-else>)))
<plan-schema> ::=  (<plan-name> [<plan-argument-value>+] [<time-
   annotation>]))
<variable-name> ::= <symbol>
<plan-schema-list> ::= <plan-schema>+
<start-time> ::= <absolute-time-point>
<end-time> ::= <absolute-time-point>
<frequency> ::= <time-measure>
<number-times-completed> ::= <number>
<max-number-of-attempts> ::= <number>
<min-time-attempt> ::= <time-measure>
<max-time-attempt> ::= <time-measure>
<plan-argument-value> ::= <value-description>
```

```
<return-expression> ::= (RETURN <return-value>)
<return-value> ::= <single-value> | <plan-schema> | <every-type-of-
  variable>
<every-type-of-variable> ::=  <variable-plan-schema>|
  <variable-plan-schema-list> |
  <variable-name> |
  <variable-domain-dependent-time-unit> |
  <variable-domain-dependent-time-point> |
  <variable-domain-dependent-time-interval>|
  <variable-domain-dependent-time-shift>
  <variable-cyclical-time-points>|
  <variable-cyclical-time-annotations>
```

APPENDIX B: EXAMPLE: A GESTATIONAL DIABETES MELLITUS (GDM) GUIDELINE IN ASBRU

The following represents a part of a guideline used in the "Sweet Success" California Diabetes and Pregnancy Program (CDAPP) at Stanford University Medical Center [CDAPP, 1992] for controlled observation and treatment of noninsulin-dependent gestational diabetes mellitus (GDM type II): first, the verbal form (an excerpt from the CDAPP); second, in a graphical version; and third, in the ASBRU language.

B.1: VERBAL VERSION

Implicit or not mentioned intentions and conditions in the guideline below have been acquired from domain experts and appear in the Asbru representation of the example.

Observation and Treatment of Gestational Diabetes Mellitus (GDM)

GLUCOSE MONITORING:
(after GDM was dedected in third trimester pregnancy, tested by a glucose tolerance test (GTT) being between 140 and 200 mg/dl)
(1) Patients will check glucose values four times/day (fasting and one hour postprandial glucose)
(2) Preprandial, bedtime and 2 AM blood glucose will be added at the discretion of the physician
(3) ... deleted ...
(4) Treatment goals should be no higher than 130 mg/dl for 1-hour post meals, < 100 mg/dl fasting and preprandial
(5) ... deleted ...

NUTRITION:
(1) Patients should be taught a diet based on the patients' weight, activity level and number of fetuses (regular meals: 3 meals, 3 snacks).
... (omitted for lack of space) ...
INSULIN THERAPY:
... (omitted for lack of space) ...

B.2: GRAPHICAL VERSION

Figure 3.5 is a zoom of Figure 3.4.

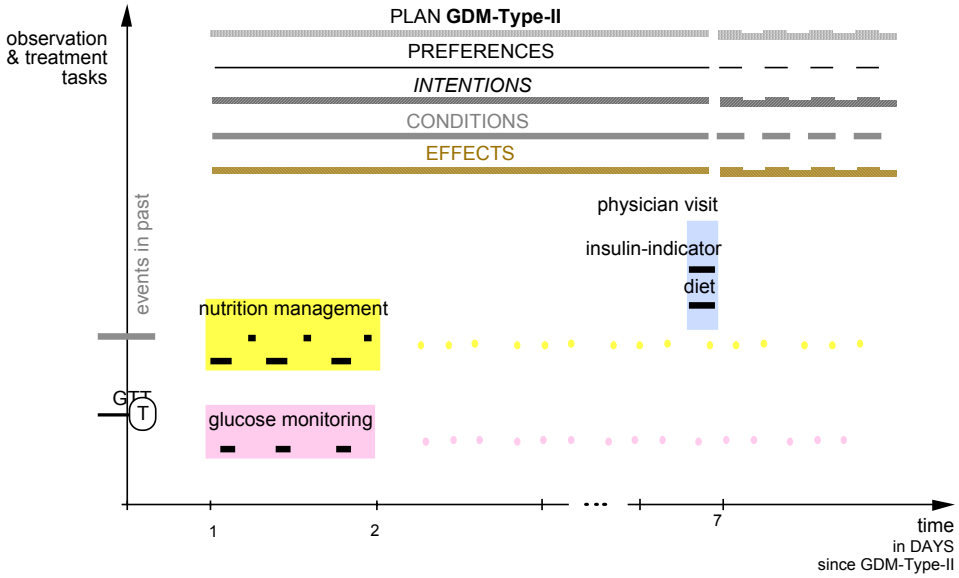

Figure 3.5 Graphical version of a part of the guideline, used in the at the CDAPP program for controlled observation and treatment of GDM Type II.

B.3: ASBRU LANGUAGE

The plan body consists of three plans that are executed in parallel. These plans are decomposable into other plans, which exist in the guideline-specification library. Nondecomposable plans are executed by the executing agent. Plan names are written in bold characters.

```
(PLAN observing-GDM-Type-II
;; the following time-annotations are local to the GDM example
(DOMAIN-DEPENDENT TIME-ASSIGNMENT
     (SHIFTS DELIVERY <- 38 WEEKS)
     ;; time shift from CONCEPTION
     (POINT CONCEPTION <- (ask (ARG "what is the conception-date?")))
(ABSTRACTION-ASSIGNMENT
     (CYCLICAL
        MIDNIGHTS <- [0, 0 HOURS, 24 HOURS]
        BREAKFAST-START-TIME <- [0, 7 HOURS, 24 HOURS]))
```

```
(PREFERENCES
     (SELECT-METHOD EXACT-FIT)
     (START-CONDITION AUTOMATIC))
  ;; The match in the filter conditions needs to be exact and the plan
  starts as soon as it is in a ready state


(INTENTION:INTERMEDIATE-STATE
  (MAINTAIN blood-glucose-post-meal (<= 130) * ;; a raw data value (no context)
    [[24 WEEKS, 24 WEEKS], [DELIVERY, DELIVERY], [_,_], CONCEPTION])
  (MAINTAIN blood-glucose-fasting (<= 100) *   ;; a raw data value (no context)
    [[24 WEEKS, 24 WEEKS], [DELIVERY, DELIVERY], [_,_], CONCEPTION])
  (MAINTAIN STATE(mothers-body-weight-gain)
    (OR SLIGHTLY-LOW NORMAL SLIGHTLY-HIGH) GDM-Type-II ;; a context-specific value
    [[24 WEEKS, 24 WEEKS], [DELIVERY, DELIVERY], [_,_], CONCEPTION])
)
(INTENTION:INTERMEDIATE-ACTION
  (MAINTAIN diet regular-meals GDM-Type-II
    [[24 WEEKS, 24 WEEKS], [DELIVERY, DELIVERY], [_,_], CONCEPTION])
);; maintain a diet with regualr meals



(INTENTION:OVERALL-STATE
  (AVOIDED STATE(blood-glucose) HIGH GDM-Type-II
    [[24 WEEKS, 24 WEEKS], [DELIVERY, DELIVERY], [7 DAYS,_], CONCEPTION])
) ;; avoid high blood-glucose level (as defined in the context of therapy
  for GDM type II)
    ;; for more than 7 days, in the period from 24 conception weeks to
  delivery,
    ;; using the estimated conception date as a reference point
(INTENTION:OVERALL-ACTION
  (MAINTAIN visit-dietitian regularly GDM-Type-II
    [[24 WEEKS, 24 WEEKS], [DELIVERY, DELIVERY], [3 MONTHS,_],CONCEPTION])
) ;;patient had visited dietitian regularly for at least three months


(SETUP-PRECONDITIONS
  (PLAN-STATE one-hour-GTT COMPLETED
    [[24 WEEKS, 24 WEEKS], [26 WEEKS, 26 WEEKS], [_,_], CONCEPTION])
) ;; The patient must have had a glucose-tolerane test (another plan in the
  library) successfully completed
(FILTER-PRECONDITIONS
  (one-hour-GTT (140, 200) pregnancy
    [24 WEEKS, 24 WEEKS], [26 WEEKS, 26 WEEKS], [_,_], CONCEPTION])
)
(SUSPEND-CONDITIONS
  (STATE(blood-glucose) HIGH GDM-Type-II
    [[24 WEEKS,24 WEEKS], [DELIVERY, DELIVERY], [4 DAYS,_], CONCEPTION]
  ;; suspend if high blood-glucose level (in the context of GDM type II therapy) exists for at least 4 days
    (SAMPLING-FREQUENCY 24 HOURS))
)
(ABORT-CONDITIONS (OR ACTIVATED SUSPENDED)
  (insulin-indicator-conditions TRUE GDM-Type-II *
    (SAMPLING-FREQUENCY 24 HOURS))
)
(COMPLETE-CONDITIONS
  (delivery TRUE GDM-Type-II * (SAMPLING-FREQUENCY 30 MINUTES))
)
```

```
(RESTART-CONDITIONS ;; restart from a suspended state
   (STATE(blood-glucose) (OR NORMAL SLIGHTLY-HIGH) GDM-Type-II
     [[24 WEEKS,24 WEEKS], [DELIVERY, DELIVERY], [_,_], CONCEPTION]
     (SAMPLING-FREQUENCY 24 HOURS))
)

(PLAN-EFFECTS (GDM-Type-II glucose NORMAL
              ([_, _], [_, _], [10 MINUTES, 60 MINUTES], *NOW*) 0.97)
)
(DO-ALL-TOGETHER
   (glucose-monitoring)
   (nutrition-management)
   (observe-insulin-indicators)
));; the plan body is a concurrent one and comprises three plans that start
   together, all of which need to complete

(PLAN glucose-monitoring
   (PREFERENCES
     (SELECT-METHOD EXACT-FIT)
     (START-CONDITION AUTOMATIC)
   )
   (INTENTION:INTERMEDIATE-STATE
     (MAINTAIN STATE(blood-glucose)
           (OR NORMAL SLIGHTLY-HIGH) GDM-Type-II
           [[24 WEEKS, 24 WEEKS], [26 WEEKS, 26 WEEKS], [_,_], CONCEPTION])
   )
   (SETUP-PRECONDITIONS
        (glycometer-equipment-available TRUE GDM-Type-II *)
   )
   (FILTER-PRECONDITIONS
     (GDM-Type-II-diagnose TRUE pregnancy *)
   )
   (DO-ALL-TOGETHER
     (monitor-fasting-glucose (ARG NORMAL glucometer))
     (monitor-one-hour-after-breakfast-glucose
                                (ARG NORMAL glucometer))
     (monitor-one-hour-after-lunch-glucose (ARG NORMAL glucometer))
     (monitor-one-hour-after-dinner-glucose (ARG NORMAL glucometer))
     (IF (physician-decided-more-analyses TRUE GDM-Type-II *)
          THEN  (monitor-alternative-times
                                    (ARG NORMAL glucometer))
   )))

(PLAN monitor-fasting-glucose (ARG glucose-value device)
   (PREFERENCES
     (START-CONDITION AUTOMATIC))
   (EVERY
     (START (FIRST(MIDNIGHT) after (ACTIVATED  *self*))
     ;; first midnight after started (activated) the current plan
     (TIME-BASE [[-1 HOURS, -1 HOURS], [-10 MINUTES, -10 MINUTES], [_,_],
                 BREAKFAST-START-TIME])
     (UNTIL (COUNT-APPEARANCE 3
              (blood-glucose STATE(blood-glucose) HIGH GDM-Type-II
               [[24 WEEKS,24 WEEKS], [DELIVERY, DELIVERY],
                 [3 DAYS,7 DAYS], CONCEPTION] ))
     ;;elevated-blood-glucose more than 3 times
     (observe blood-glucose device glucose-value)
   END-EVERY )
```