# JTF ATD Core Plan Representation

**R. Adam Pease**
Teknowledge
1810 Embarcadero Rd
Palo Alto, CA 94303
apease@teknowledge.com

**Capt. Todd M. Carrico**
Armstrong Lab
OL AL HSC/HRGO, Bldg. 190; 2698 G. St.,
Wright-Patterson AFB, OH 45433
tcarrico@alhrg.wpafb.af.mil

## Abstract

The Core Plan Representation (CPR) [Pease and Carrico, 1996] is an effort to develop a plan representation which supports the representation needs of many different planning systems. It is being developed for the DARPA Joint Task Force Advanced Technology Demonstration (JTF ATD) [Hayes-Roth, 1995], [Hayes-Roth & Erman, 1994], [Carrico, 1996]. The goal of this effort is to leverage common functionality and facilitate the reuse and sharing of information between a variety of planning and control systems. The CPR embodies a standard which is general enough to cover a spectrum of domains from planning and process management to workflow and activity models. In addition, the representation can support complex, hierarchical plan structures. The prime motivation for the CPR effort is to address plan interchange requirements of several military planning systems, but this effort goes beyond military planning and presents a more general plan representation.

## Background

The design of the CPR is an attempt to unify the major concepts and advancements in plan and process representation into one comprehensive model. This section first provides some motivation for the CPR effort. Next, a significant design consideration is explained. References are then given for research efforts which had particular influence in the design process.

### Motivation

There are two significant payoffs to the CPR effort. The first is that creation of a base plan representation will facilitate information interchange among different planning systems. Imagine a generic military planning situation. A crisis develops and a joint task force is formed. The leadership and staff use a planning application to develop guidance for their subordinate commands. This guidance includes background on the situation, objectives which must be met to contain the crisis, constraints on the actions of the task force and high level specification of the schedule of operation. This information is passed to individual commands which have specific requirements and methods of planning. A standard plan framework enables improved information transfer to these specialized planning applications. Continuing to follow this generic and hypothetical example, the commander of the air component of the task force and his staff will use their superiors' objectives to develop more detailed objectives, lists of targets which support those objectives, and then repeatedly create a schedule of aircraft sorties to destroy those targets. Pilots flying those sorties could benefit from performing simulated runs. A core plan representation enables information from the plan to be transferred to simulation entities possibly allowing a single pilot to fly along side computer generated forces simulating the other pilots in his flight. While information transfers of this sort will rarely be complete, and will often require further augmentation and elaboration for the new application, the CPR will reduce the amount of manual rekeying and reformulation of existing data.

The second payoff is in the creation of common services based on the CPR. There are two broad areas of services with immediate utility. The first area is visualization. Manufacturing, business planning and construction management all share several basic forms of visualizing plan information. The CPR enables creation of these common views which will dramatically reduce implementation time for specific systems. Well designed common viewers can then be specialized for particular planning applications instead of written from scratch. The second area is scheduling. Many important advances have occurred in the operations research and artificial intelligence communities which allow software systems to provide significant aid in complex scheduling problems. A complete scheduling system can rarely be build on generic techniques alone, but the CPR enables the creation of generic scheduling tools which eliminate the need to build

these tools from scratch each time a new scheduling domain is targeted.

## A Design Objective

The CPR attempts to model the "basic level" [Rosch, 1976] of plan representation for the given domain of planning needs. The basic psychological level is a simple yet elegant concept. In an ontology of several levels describing a given domain, there is one level at which humans associate the largest amount of information. For example,

| | | |
|---|---|---|
| **SUPERORDINATE** | ANIMAL | FURNITURE |
| **BASIC** | DOG | CHAIR |
| **SUBORDINATE** | RETRIEVER | ROCKER |

**Table 1 : Example from [Rosch, 1976]**

There are several attributes of the basic level. One is that this is the level at which terms are used in neutral contexts. For example [again after Rosch] *There is a dog on the porch* as opposed to *There's a mammal on the porch* or *There's a wire-haired terrier on the porch.* The latter two sentences are unusual and require further explanation.

For plan information, simply specifying objects Entity and Relationship is not enough. Nor would defining Plan as an object without attributes be sufficient. On the other hand, specifying the Plan as having weatherForcecast attribute, or defining a NonshareableResourceConstraint would be too specific. Few plans require this sort of information. The CPR specifies plan information at the lowest level common to all planning systems of interest.

It should be noted that the basic level is specific to the domain being covered. If the context of interest is air campaign planning then a weather forecast object is useful and the CPR must be specialized for that context The CPR may be specialized to cover more restricted domains in greater detail. The basic level of plan information for air campaign planning will container a much richer set of information than is common to planning in general.

The basic level provides an initial guide for the inclusion of concepts in the CPR. It should be noted that all basic level objects need not be included nor must all objects meet the basic level criterion. Practicality and relevance are important and may outweigh the basic level criterion.

## Prior research

This work has been influenced by many other efforts, a review of which is outside the scope of this document. The interested reader is referred to [Tate et al, 1990], [Allen et al, 1990], and [Zweben and Fox, 1994].

However, several efforts in particular have influenced the CPR. The Knowledge Representation Source Language (KRSL) plan representation bears many similarities to CPR. Several individuals who developed the KRSL plan representation have contributed significantly to the CPR design. The goals of the two efforts are slightly different however. While KRSL is working on developing an ontology of plans and activities, CPR is striving for an object oriented software design developed with a strong ontological awareness. It should be noted that KRSL like CPR is also an ongoing effort although it began much earlier. I-N-OVA [Tate, 1996] is another ongoing effort which bears some similarity to CPR due to the significant contribution of Austin Tate to the CPR effort. Another important related effort is the Process Interchange Format [Lee, 1996]

# Building the CPR

To develop an appreciation for the CPR design, some of the component objects will first be presented. The plan representation will be constructed step by step in order to capture the design motivations, considerations, and open design issues. Finally, all the components will be brought together to form the complete CPR. Readers who are only interested in the finished design may skip this section.

## Foundation Concepts

The first step is to identify the minimal concepts necessary to represent any plan. This draws on the work of the Knowledge Representation Source Language, the DARPA Plan Ontology Construction Group, and O-Plan [Tate et al, 1996]. The initial concepts are *Action*, *Actor*, *Objective*, and *Resource*. An *Action* is performed by an *Actor*. The motivation behind performing the *Action* is to accomplish some *Objective*. In performing the *Action*, the *Actor* may utilize a *Resource*. An *Actor* in one *Action* could be a *Resource* in another.

The minimal set is shown in figure 1. This is the set of core concepts from which the CPR will be constructed.

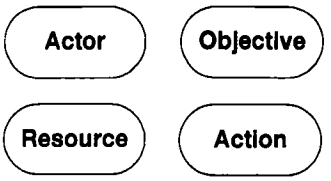



**Figure 1 - Basic Concepts**

## From Concepts to Design

The basic design can be created from the above concepts. A *Plan* consists of one or more *Actions* performed in pursuit of some *Objective*. As noted above, *Actors* may use *Resources* in the performance an *Action*. *Actor* and *Resources* are not directly elements of the *Plan*, but rather of the *Action* itself.

*Plans* require specifications of time and space. *TimePoint* represents some association of the *Action* with time. Though a number of *TimePoints* may provide information about an *Action*, every *Action* has at least a beginning and end, though either may be infinite or periodic. *SpatialPoint* may represent an exact location, or a vague area. It may ground the objects both in absolute and relative terms. *SpatialPoints* are also associated with *Actions*. Every *Action* takes place in some space whether the space is real or virtual.

*Constraints* need to be added our initial set of classes. *Constraints* are restrictions on other elements of the plan. For example, we might need to specify that one *Action* must take place in a certain proximity to another *Action*, or that the end *TimePoint* of one *Action* occurs before the begin *TimePoint* of another. *Constraint* has no single place in the CPR where it would be most appropriate. As such, it is left as an object which can be contained in any object of the plan.

It is clear that from this basic framework that a host of other required elements could be identified and added. Because the CPR was meant to be a general purpose, flexible representation, an extremely complex or constraining design is inappropriate. To that end, elements are carefully considered for both relevance and generality before being added.

Due to aggregate nature of plans, it is appropriate to allow a *Plan* to be associated with another *Plan*, where one would be the parent plan and the other a sub-plan. A similar argument may be made for both *Objectives* and *Actions*, representing sub-objectives and sub-actions respectively.

During execution of the *Plan*, *Objectives* are reviewed in order to gauge the effectiveness of the *Plan* and identify when the *Plan* is complete. This review is performed against a set of evaluation criteria relevant to the Objective. The entity EvaluationCriterion is added to *Objective*. The entity *EvaluationCriterion* is added to address this need.

This design at this point is shown in Figure 2.

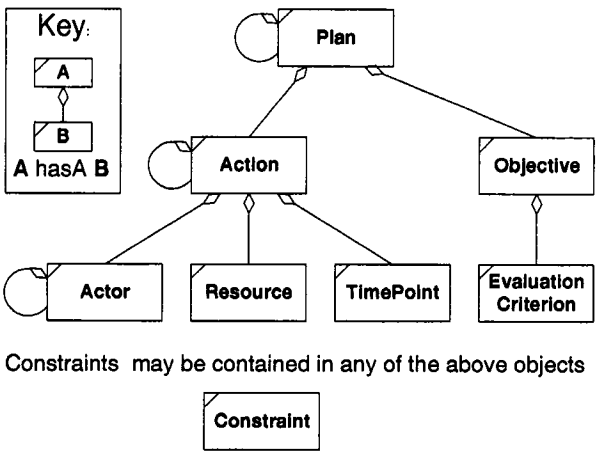



**Figure 2 - CPR framework**

## Completing the Design

This section continues explaining the design by adding in attributes to the classes. Attributes which are plural may be understood as containing a list of objects.

First, *Plan* needs to be completed. It already contains *subPlans*, *actions*, and *objectives*. *Plan* must have a set of metadata describing the merits of one plan relative to others. This is contained in the attribute *evaluation*. It must be noted that this is different than an *EvaluationCriterion*. An evaluation of a *Plan* is an assessment of how well constructed it is, or how applicable it is to the current situation. An *EvaluationCriterion* specifies how a planner may assess whether a given *Objective* has been met in the current state of the world.

It is also valuable to allow metadata about the status of plan creation. For this purpose, the *issues* attribute is added. Finally, alternative plans, if available, should be accessible from the current plan. To support this, the attribute *alternative* is defined as a list of pointers to *Plans*.

*Action* may now be elaborated. We have already identified the *subActions*, *actors*, *resources*, and *begin* and *end* attributes. *DomainObjects* are defined to represent entities referred to in an *Action* which are not the *Actor* or a *Resource*. For example, the recipient of a mail message would be recorded as an instance of *DomainObject*.

*Plans* must be able to represent incomplete information about the world. The classes *Uncertainty* and *Imprecision*

represent two kinds of incomplete information. *Uncertainty* captures the degree of confidence in information [Feller, 1957]. *Imprecision* captures the degree to which an exact value cannot be specified [Zadeh, 1978]. *Uncertainty* and *Imprecision* are both required but different plans may have very different requirements for handling them. Both objects are currently left undefined.

It is clear that most plan objects require a *name*. Any class in the *Plan* may also have an associated *Uncertainty* or *Imprecision*. In addition, it is valuable to be able to add *Annotations* to any element of the plan. As mentioned above, *Constraints* may also apply to any plan object. For this reason, the common superclass of *PlanObject* is created containing these attributes.

*Actors* may have objectives of their own and so a reference to the plan objectives is added and called *objectives*. Since the *Actor* may not be a single person, but represent an aggregate of some kind, the attribute *subActors* is added.

*DomainObject* is intended to contain information about an entity which is referred to in the description of an *Action* which is not an *Actor* or a *Resource*. For example, the recipient of an email message would be a *DomainObject* and the sender would be an *Actor*. This object is helpful in creating a correspondence with the design of PIF [Lee, 1996].

An *Agent* is an object which allows a correspondence among *Actors*, *Resources*, and *DomainObjects* when those objects describe different roles which are filled by the same real world entity.

*Objective* contains *type*, *subObjectives*, and *evaluationCriteria*. *Value* is added to hold the actual objective element. *Objective* also references the list of *Actions* in the *Plan* which are meant to satisfy it.

In order to complete *Constraint*, additional fields are added to hold the terms of the constraint expression and any subordinate constraints. These are represented by the attributes of *term* and *subConstraint*.

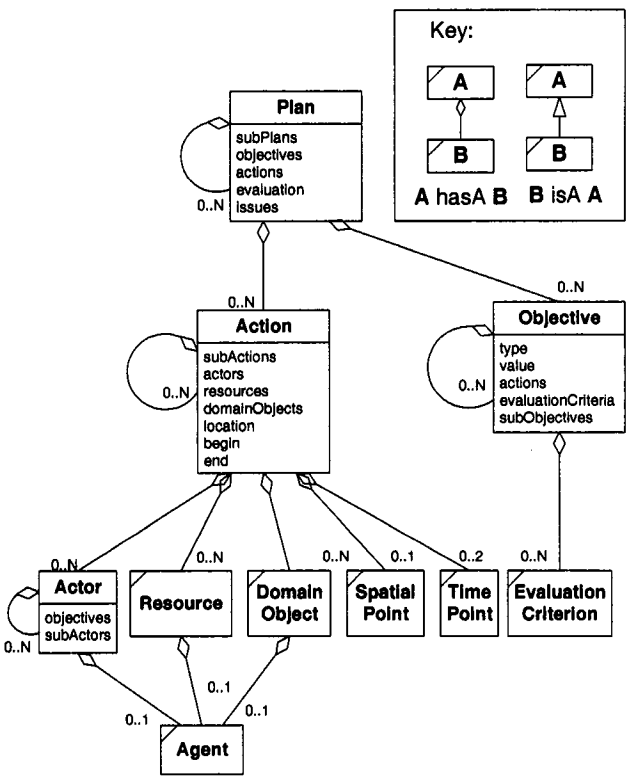
An *Annotation* is designed to hold any type of unstructured data, paragraphs of text, pictures, video etc. It exists to hold any information which supports understanding or maintenance of the plan but it not strictly part of the plan itself. *Annotations* consist of a *name* and the body of the annotation which is left unspecified. In addition, the annotations could be constructed hierarchically to form linked documentation, and thus a list of *subAnnotations* is required.

Finally, some additional components were identified, but require additional consideration. These classes capture *Annotations* which are more structured than the free text or image that *Annotation* can handle. Each defines a specialization which is still very generic. The first is *Fact*, which is a basic data component having a *name* and a list of *terms*. Another is a *Frame*, which consists of a set of attribute-value pairs given as the attribute *slotList*. Also defined is *Rule*, consisting of a *precondition* and *postcondition*. It is designed to hold the kind of information which might be contained in an expert system rule. Lastly we have *Assumption* which is designed to record information which gives the information which supports the creation of a given instance of a PlanObject.

## The complete design

The CPR design addresses the level of highest possible value, utility and commonality for information interchange and the development of common services. Many details must be left unspecified because committing to a specific design detail would prevent common description between different planning applications.

In most design efforts decisions and tradeoffs are made and not recorded. Subsequent design efforts do not have the advantage of reusing or being influenced by the designs which were not chosen as well as the positive products of a design. Each new design must largely start from scratch. The CPR effort has taken the opposite approach. Whenever reasonable choices are encountered, they are recorded (see [Pease and Carrico, 1996]).



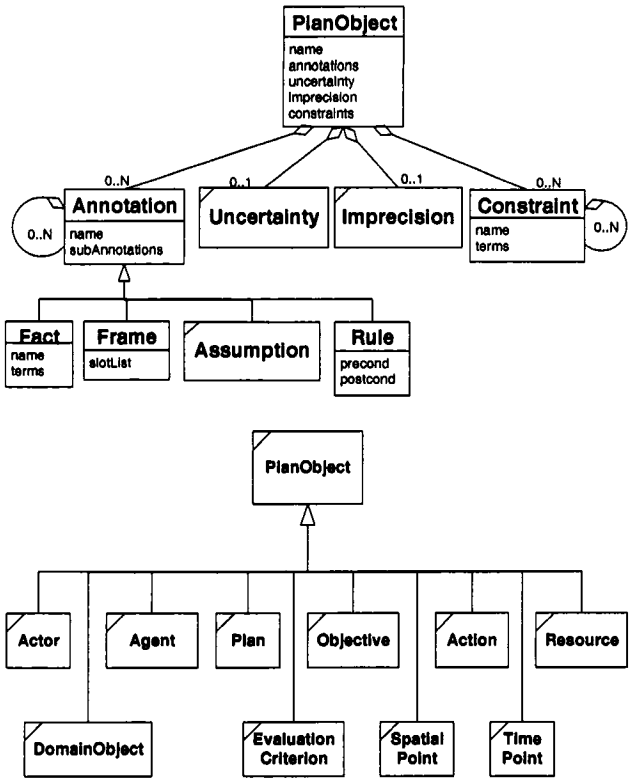



**Figure 3 : The CPR Design**

## Conclusion

This proposal has provided references to planning research and suggested the need for a common, unified representation of a plan. Basic plan concepts were offered, concepts were assembled into a design framework and then refined.

Comments, concerns, or deficiencies should be submitted to Adam Pease at apease@teknowledge.com.

## Acknowledgements

Austin Tate has provided extensive advice and criticism throughout the development of the CPR. Thanks are also due to the CPR meeting attendees from July 15-16, 1996, and members of the ARPA Plan Ontology Construction Group. Rick Hayes-Roth brought his vision of the JTF Schema together with many years of experience to provide valuable guidance and input. As chief architect of the JTF ATD, his unique perspective on both design and implementation proved invaluable.

Special thanks are due to John Schill and Lou Hoebel whose financial support and encouragement made this work possible

## References

Allen, J., Hendler, J., and Tate, A., 1990. *Readings in Planning*, Morgan Kaufman.

Carrico, T., 1996. The Object Model Working Group C3I Schema. Armstrong Lab Technical Report, limited distribution.

Feller, W., 1957, *An Introduction to Probability Theory and its Applications*, John Wiley & Sons.

Hayes-Roth, F. and L. Erman. 1994. The Joint Task Force Architecture Specification (JTFAS). Palo Alto, CA: Teknowledge Federal Systems.

Hayes-Roth, F. 1995. The JTF ATD architecture for agile, mobile, collaborative crisis response: A case study in architecture-driven development. In Proceedings of the First International Workshop on Software Architecture. Pittsburgh, PA: Carnegie-Mellon Univ.

Lee, J., (editor), Grunninger, M., Jin, Y., Malone, Tate, A., Yost, G., and other members of the PIF Working Group. 1996. The PIF Process Interchange and Framework Version 1.1, MIT Center for Coordination Science, Working Paper #194.

Pease, A., Carrico, T., 1996. The JTF ATD Core Plan Representation: Request for Comment. Armstrong Lab ASC 96-2489 WU 29500043 - AL/HR-TP-96-9631.

Rosch, E., Mervis, C., Gray, W., Johnson, D., and Boyes-Braem, P., 1976. Basic Objects in Natural Categories, *Cognitive Psychology* 8:382-439.

Tate, A., Hendler, J., and Drummond, M., 1990. A Review of AI Planning Techniques. In *Readings in Planning*, pp. 26-49, (Allen, Tate and Hendler, eds), Morgan Kaufmann.

Tate, A., 1996. Representing Plans as a Set of Constraints - The <I-N-OVA> Model. In *Proceedings of the Third International Conference on Planning Systems* ed. B. Drabble, AAAI Press.

Tate, A., Drabble, B., Dalton, J., 1996. Oplan: A Knowledge-Based Planner and its Application to Logistics. In *Advanced Planning Technology*, ed. A. Tate, AAAI Press, pp 259-266.

Zadeh, L., 1978. Fuzzy Sets as a Basis for a Theory of Possibility, *Fuzzy Sets and Systems*, Vol 1, No. 1, pp 3-28.

Zweben, M., and Fox, M., 1994. *Intelligent Scheduling*, Morgan Kaufman.