THE AIR CAMPAIGN PLANNING KNOWLEDGE BASE

Version 4.2

Thomas J. Lee
Artificial Intelligence Center
SRI International

October 21, 1999

CONTENTS

1 INTRODUCTION

This document describes the Air Campaign Planning Knowledge Base (ACP KB), a knowledge base developed at SRI for the domain of military air campaign planning. A *knowledge base* contains knowledge in a form that is useful for solving problems in a particular domain.It is used in conjunction with SIPE-2[*], a domain independent, state-of-the-practice artificial intelligence (AI) planning system.

SIPE-2 has been applied to a number of military domains. Under the DARPA Joint Forces Air Component Commander (JFACC) program, it was used as the generative planning agent in the Continuous Planning and Execution Framework (CPEF) system developed at SRI [Myers 1998**]**. The ACP KB described herein (aside from a few subsequent extensions) is the knowledge base that was used to demonstrate CPEF.

The ACP KB has been used in numerous research programs. Under the DARPA/RL Planning Initiative (ARPI) program, it was used as the nowledge base in the generative planning agent within the Multiagent Planning Architecture (MPA) in the Technology Integration Experiment 97-1 (TIE 97-1).An earlier version of the ACP KB was used in 1996 for ARPI's Fourth Integrated Feasibility Demonstration (IFD-4).

[Bienkowski 1997] provides more details on SIPE-2's role in IFD-4. [Wilkins 1997] gives details on running SIPE-2. SIPE-2 in JFACC and TIE 97-1 relied on the Advisable Planner module, which is described by [Myers 1996].

This report is organized as follows. Section 2 gives an overview of the ACP KB. Section 3 describes the plans that are generated by SIPE-2 using the ACP KB. Sections 3–8 describe the components of the ACP KB. Section 4 describes the threat modeling; Section 5, the object-class hierarchy; Section 6, the goals and operators; Section 7, the deductive rules; and Section 8, the predicates. Section 9 lists acknowledgments. Section 10 lists the references cited in the text.

2 ACP KB OVERVIEW AND SCENARIO

The ACP KB encodes knowledge for automatically generating that portion of an air campaign plan whose purpose is to achieve *air superiority.* That is, SIPE-2 uses the ACP KB to generate a set of primitive actions, executable by a group of aircraft or other assets, that provide protection to friendly aircraft and centers of gravity (COGs)[†] from all enemy threats enumerated in the scenario.

The ACP KB fulfills two roles - *strategy selection* and *support requirements generation*. Strategy selection includes selecting defensive schemes to protect COGs in friendly territory, and selecting targets to attack. Support requirements generation consists of determining the support

---

[*] SIPE-2: System for Integrated Planning and Scheduling. SIPE-2 is a trademark of SRI International. All product or company names mentioned in this document are the trademarks of their respective holders.

[†] A center of gravity represents a target or target set whose destruction would have a far-reaching impact (i.e., impact beyond the loss of a capability provided by that target).

required by all actions chosen during strategy selection, and selecting the primitive actions needed to provide that support. Strategy selection and support requirements generation are run in separate stages.

Between these two stages, an external program is run, which takes the targets from the first part, supplements each with additional information, *packages* the results into groups of targets to be attacked together, and schedules the targets in accordance with aircraft availablity. Historically, the program used was CTEM; for details, see [Lee 1996]. CTEM's output is used as input to complete the second part of the plan for SIPE-2. Internally, these two parts are represented as two separate subplans. These subplans are referred to the *pre-CTEM* and *post-CTEM plans*.

In many demonstrations, only strategy selection is of interest. In that case, only the strategy selection stage is run; packaging, scheduling, and computing support requirements are omitted, and only the pre-CTEM plan is generated.

The plan generation process takes a *scenario* as input. The scenario describes in detail the situation to which the generated plan is responding. It includes representations of geography; a target database; target networks (i.e., a model of how targets function together and interact); friendly COGs; and friendly air assets and airbases, as well as an intelligence estimate of the kinds and degrees of the threats posed by the enemy to air superiority.

The ACP KB has been implemented and demonstrated in the Cyberland scenario, which was used for JFACC. This scenario depicts a conflict between two hypothetical countries, East Cyberland (the U.S. ally) and West Cyberland, both located on the island of New Guinea. The goal of the United States is to protect East Cyberland from invasion (including an important oil-producing site), and restore the prewar borders. West Cyberland is a substantial regional military power, with a potent air force and a reasonably effective integrated air defense system (IADS).

The main sources for the knowledge used to construct the ACP KB were interviews conducted with USAF officers familiar with various aspects of air campaign planning. This knowledge was supplemented with published material, including [Murray 1995] and [Deptula 1996].

3 PROBLEMS AND PLANNED SOLUTIONS

Real-world air campaign plans have a hierarchical structure. The ACP KB represents a subset of that structure, as follows. An *air objective* is a high-level goal attainable through the use of air power. Air objectives are decomposed into *air tasks*. Tasks are decomposed into *activities*: an attack on a particular target is an activity. Activities are decomposed into *support missions*. Each support mission occurs on a particular day and time, and represents one sortie by a group of like aircraft (e.g., F-15Cs).

Each plan generation process uses the knowledge encoded in the scenario. In addition, the pre-CTEM planning process takes as input one air objective. That air objective is to achieve air superiority in the area covered by the scenario. This goal can express the desire to attain either air superiority proper, or air supremacy; the latter implies a greater reduction of and/or stronger defense against all enemy threats.

The pre-CTEM plan (i.e., the solution generated for the air superiority planning problem) contains two kinds of primitive actions—activities for each target selected for attack, and support missions for all activities not involving direct attacks on targets.

The post-CTEM planning process takes as input groupings of these target activities into packages (grouping is done by CTEM). Its output is a set of support missions that will accomplish these attacks.

Example planning problems are contained in the source file released with the SIPE-2 distribution [Wilkins 1997] `problems.sipe`. The first three problems—`AS-A`, `AS-B`, and `AS-C` are identical problems containing the high-level air superiority goal the KB solves. Other problems are used to test portions of the KB.

4 MODELING AIR CAMPAIGN PLANNING

4.1 CAPABILITY-BASED MODELING

This section describes how various elements of a scenario are modeled in order to generate plans that, when executed, select and attack enemy capabilities in support of high-level planning goals. A key feature of the ACP KB is that it enables SIPE-2 to create plans that deal with groups of targets that work together to provide a capability to the enemy. The KB provides primitives for aggregating targets into groups, and associating with that group a capability it provides. For example, all runways at an airbase provide a TAKEOFF-LANDING network that provides a TAKEOFF-LANDING capability at that airbase. Each such group is called a *network*. Furthermore, the same primitives can be used to group networks into higher-level networks that provide higher-level capabilities. For example, the TAKEOFF-LANDING, MUNITIONS, MAINTENANCE, and AIRCRAFT capabilities of an airbase work in concert to provide the capability to conduct AIR-OPERATIONS from that airbase.

Associated with each network is an effectiveness level between 0 (inoperative) and 1 (fully capable). As network components are damaged or otherwise degraded, the effectiveness levels in the modeled networks are reduced by an amount that is a function of the criticality of the degraded component. The ACP KB contains rules for computing effectivenesses, which are reusable from scenario to scenario. Effectiveness levels are represented by SIPE-2 LEVEL predicates [Wilkins 1997].

4.2 NETWORK REPRESENTATION

Each network is uniquely identified by a *capability* and a *place*. The place can be a target, a small area, a large region, etc., as desired for modeling the range or extent of the capability.

Each network is also characterized by a *composition*. The composition determines the effectiveness of the parent network as a function of the effectivenesses of its children. There are three different compositions:

- Additive: all components contribute in proportion to their weight

- Critical: parent effectiveness is the minimum of all components
- Redundant: parent effectiveness is the maximum of all components.

These characteristics are represented using a NET predicate for the network.

A parent network has one or more components. A component may be either a target or another network, and may be a part of many networks. Like a network, a component is uniquely identified by a capability and a place. If the component is a target, then the target itself is the place.
A component also has a weight associated with it, which is used in determining the impact of degradation in the component's effectiveness on that of the parent. Each component is represented by a PROVIDES predicate.

Some networks have capabilities that are in part intrinsic: that is, even if all of its components were rendered completely ineffective, the network could still continue to function at some threshold level of effectiveness, below which it cannot be reduced. This threshold is represented by a NEEDS predicate.

Capability-based network modeling has been used, in the ACP KB, to represent IADS networks. It could also be used to represent other target networks of interest in air campaign planning, for example production networks including mining, manufacturing, and distribution components.

4.3 THREATS

The ACP KB models threats posed by surface-to-air missiles (SAMs), as well as various threats from enemy aircraft, including strike threats to ground- and sea-based friendly forces and territory, threats to friendly aircraft flying over friendly territory, and threats to friendly aircraft flying over enemy territory. The model allows for threats from antiaircraft artillery (AAA) and theater-ballistic missiles (TBMs), as these are part of the air superiority picture. (However, the current implementation of the KB ignores these threats.)

The following information is associated with each threat:

- The kind of threat
- The associated network (capability and place)
- Whether the threat is daytime-only or 24-hour
- The base level of the threat; the base level reflects an intelligence analysis of the lethality of the threat against the most likely platforms or blue COGs it could be employed against.

The current level of a threat is its base (initial) level times the current effectiveness of the associated network, reflecting any damage or other degradation due to attacks upon it. As with network effectivenesses, the current level of a threat is represented by a LEVEL predicate (although threat levels are not constrained to be in the range between 0 and 1).

Threats are represented by THREAT predicates. For example:

- `(threat strike-threat air-operations Van-Nuys daytime 160)`

- Says the ground strike threat emanating from Van Nuys airbase as a daytime-only threat and has a base level of 160.

4.4 TIME

The time granularity of the ACP KB is 24 hours. All actions are considered to take at least one full day to accomplish, for historical reasons. First, the original purpose of the ACP KB was to estimate resource requirements (i.e., the numbers of aircraft sorties) required by a plan. One-day granularity is appropriate for this estimate, but is problematic for other uses of the ACP KB. For instance, it would be useful to sequence certain actions within a 24-hour period, perhaps designating some to occur at night. The ACP KB cannot do so.

5 CLASSES AND OBJECTS

SIPE-2 includes primitives for expressing class-object hierarchies to define domain elements and groups thereof. SIPE-2 documentation refers to this as the *sort-hierarchy* [Wilkins 1997]. Each definition includes the class/object name, its parent class, and a property list.

The objects in the sort hierarchy define the symbolic arguments allowable in predicates. The classes and properties of an object can be referenced in constraints in SIPE-2 operators; this is a convenient way to express planning semantics.

Following is a description of the key classes used in the ACP KB, along with some illustrative objects. SIPE-2 allows two syntaxes for expressing hierarchies: a flat syntax, in which direct parent classes and child classes/objects are listed, and a nested syntax, which allows the specification of multiple levels of children in one syntactic structure. Both are used below.

Most of the classes listed below are in the source file `hierarchy.lisp`. Many of the geographical classes are in `geography.lisp`. An associated `.sipe` file is automatically generated for each of these files when the KB is modified. A few high-level classes are defined in `root-hierarchy.sipe`. The target database is contained in `target-objects.sipe`. This file is generated programmatically from a flat file representation of the target database.

5.1 PLACE CLASS

The place class is the root for all geographic classes. It is used extensively in operators to indicate a variable that may be bound to any place in the scenario.

```
CLASS: place
SUBCLASSES: region,sector,target,seaport,airbase,e-airbase;
END CLASS
```

5.2 REGION CLASS

A region is a large-scale area that may be subdivided into sectors.

```
(class region
 (object aor)
 (class land-region
       ;; Scenario-specific land-region objects are defined here
                          (object EAST-CYBERLAND (side BLUE)
       (object WEST-CYBERLAND (side RED))
       ;; These are used only to aggregate various threats into one capability
       (object northern-wc (side RED))
       (object southern-wc (side RED))
       (object eastern-wc (side RED))
       (object western-wc (side RED))
       )
(class sea-region
       ;; Scenario-specific sea-region objects are defined here
       (object pacific-ocean (side GRAY))
       (object coral-sea (side GRAY))
       )
)
```

5.3 SECTOR CLASS

A sector is an area smaller than a region.

```
(class sector
 (class land-sector
              ;; Scenario-specific sea-sectors are defined here
              )

(class sea-sector (side GRAY)
       ;; Scenario-specific sea-sectors are defined here
 )
)
```

5.4 TARGET CLASS

The target class groups targets into types and subtypes according to a classification system used by the U.S. Air Force. Each type of target has a *catcode* associated with it. Each target in the target database has one of these as a parent class.

```
(class target
 (catcode 99999)
 (object NONTARGET)
 (class movable )

(class ab-targets (catcode 71000)
       (class ab-shelter (catcode 71100)
       (class ab-ac-in-open (catcode 71200))
       (class ab-fuel-storage (catcode 71300))
       (class ab-munition-storage (catcode 71400))
       (class ab-maintenance (catcode 71500))
```

```
       (class ab-runways (catcode 71600))
       (class ab-housing (catcode 71700)))
(class c3-targets (catcode 72000
       (class c3-hq (catcode 72100))
       (class c3-leadership (catcode 72300))
       (class c3-center (catcode 72400))
       (class c3-air-defense (catcode 72500))
       (class c3-cb (catcode 72600))
       (class c3-military (catcode 72700))
       (class c3-telephone (catcode 72800))
       (class c3-satellite (catcode 72900)))
 (class power-facilities (catcode 73000)
       (class pf-nuclear (catcode 73100))
       (class pf-electrical (catcode 73200))
       (class pf-transformer (catcode 73300))
       (class pf-relay (catcode 73400))
       (class pf-refinery (catcode 73500))
       (class pf-petroleum-storage (catcode 73600))
       (class pf-pipeline (catcode 73700)))
(class storage (catcode 74000)
       (class sf-sam-ammo (catcode 74100))
       (class sf-other-ammo (catcode 74200))
       (class sf-supply (catcode 74300))
       (class sf-vehicle (catcode 74400))
       (class sf-food-ammo (catcode 74500)))
(class seaport (catcode 75000)
       (class sp-dock (catcode 75100))
       (class sp-storage (catcode 75200))
       (class sp-surface-berth (catcode 75300))
       (class sp-sub-berth (catcode 75400))
       (class sp-offload (catcode 75500)))
(class iads-targets (catcode 76000)
       (class ia-ewgci (catcode 76100))
       (class ia-sam)
              (class ia-ssam (catcode 76200))
              (class ia-tsam (catcode 76300))))
(class transportation-infrastructure (catcode (77000))
       (class ti-bridge (catcode 77100))
       (class ti-constriction (catcode 77200))
       (class ti-cargo (catcode 77300))
       (class ti-sewage (catcode 77400))
       (class ti-water (catcode 77500)))
(class marshalling-area (catcode 79000)
       (class ma-barracks (catcode 79100))
       (class ma-open (catcode 79200))
       (class ma-armor (catcode 79300))
       (class ma-artillery (catcode 79400))
       (class ma-vehicles (catcode 79500))
       (class ma-engineering (catcode 79600)))
(class wmd (catcode 80000)
       (class wmd-production (catcode 80100))
       (class wmd-storage (catcode 80200))
```

```
       (class wmd-research (catcode 80300)))
(class terrorist-camp (catcode 81000))
(class ballistic-missile (catcode 82000)
       (class bm-storage (catcode 82100))
       (class bm-production (catcode 82200))
       (class bm-launcher (catcode 82300)))
(class ground-vehicle (catcode 83000)
       (class gv-vehicle (catcode 83100))
       (class gv-artillery (catcode 83200))
       (class gv-howitzer (catcode 83300))
       (class gv-mrl (catcode 83400)))
)
```

The target database consists of a set of objects of class target. Each target is uniquely identified by a number called a Basic Encyclopedia (BE) number. Target objects are created programmatically from a flat file representation of the target database. Following is an example target object.

```
OBJECT: Sentani-Airbase-9SEN000012-71300
 PARENT-CLASS: AB-FUEL-STORAGE
 PROPERTIES:
  CATCODE = 71300,
  BEN = 9SEN000012,
  SECTOR = SENTANI-SECTOR,
  LAT = 234,
  LON = 14031,
  LATBOX = 200,
  LONBOX = 14000;
END OBJECT
```

5.5 RATING CLASS

The rating class is used as a general-purpose rating convention in numerous predicates.

```
(class rating
       (object very-high (ordinal 5))
       (object high (ordinal 4))
       (object med (ordinal 3)) ;;medium is a Sipe variable name
       (object low (ordinal 2))
       (object very-low (ordinal 1))
       (object none (ordinal 0)))
```

5.6 WHEN CLASS

The when class is used to specify when an action or support mission occurs.

```
(class when
       (object DAILY)             ;for actions which are done every day
       (class dday                ;time relative to start of a plan
              (object D+0 (ordinal 0))
              (object D+1 (ordinal 1))
              (object D+2 (ordinal 2))
```

```
              (object D+3 (ordinal 3))
              (object D+4 (ordinal 4))
              (object D+5 (ordinal 5))
              (object D+6 (ordinal 6))
              (object D+7 (ordinal 7))
              (object D+8 (ordinal 8))
              (object D+9 (ordinal 9))
              (object D+10 (ordinal 10))
              (object D+11 (ordinal 11))
              (object D+12 (ordinal 12))
              (object D+13 (ordinal 13))
              ))
```

5.7 DAYNIGHT CLASS

The daynight class is used to characterize when certain threats are extant, and when certain support missions take place.

```
(class daynight
       (object 24-hour (duration 24));ie. day and night
       (object daytime (duration 12));!assume this for simplicity
       (object nighttime (duration 12)))
```

5.8 AIRFRAME CLASS

The airframe class describes the types of friendly aircraft used in planning support missions (the capabilities of enemy aircraft are modeled more abstractly by THREAT predicates). Subclasses and properties are used to determine the suitability of aircraft for various support missions.

```
(class airframe
       (air-self-protect 0)       ;how well they handle enemy fighters
       (burn-rate 10000)          ;!just a guess
       (cas 0)                    ;appropriateness for CAS role
       (harm 0)                   ;#of AGM-88 (HARM) missiles
       (intercept 0)              ;appropriateness for air sup. role
       ;;Jamming is expressed in angular coverage of a pair of aircraft
       (jam-comm 0)               ;communications jamming
       (jam-radar 0)              ;radar jamming
       (max-offload 0)            ;tanker’s fuel
       (max-sortie-duration 8)    ;max hrs aloft, on avg.
       (radius 500)               ;combat radius for typical profile
       (service USAF)             ;default is Air Force
       (speed 250)                ;nominal/cruising speed
       (stealth 0)                ;stealth/nonstealth flag
       (class no_airframe         ;just so NO-AIRFRAME inherits props.
       (object NO-AIRFRAME));for self-delivered munitions
       (class unmanned
         (burn-rate 0)            ;unmanned airframes don’t refuel
         (object tomahawk         ;cruise missile - an airframe to CTEM
              (radius 1500)))     ;!just a guess
```

```
(class manned
(class fighter             ;ie. high performance ftrs
       (cas 5)             ;just a guess
       (burn-rate 8000);fuel/hour used, on avg. (2 engines)
       (object f-117a (stealth 1)
       (radius 600)
            (speed 420))
     (class interceptor    ;can’t drop bombs
           (air-self-protect 9)
           (cas 0)
           (intercept 9)
           (object f-14d (service USN)
                       (radius 600)
                       (speed 480))
           (object f-15c
                       (radius 860) ;profile-dependent
                       (speed 500))
           )
     (object f-15e
            (radius 1000)
            (speed 540)
            (air-self-protect 7)
            (intercept 5)) ;Usable for intercept missions
     (object f-16c;F-16s vary greatly in capabilities
            (radius 500)
            (speed 420)
            (air-self-protect 5)
            (intercept 7)  ;Used in air sup. role sometimes
            (burn-rate 4000) ;F-16 is single-engine
            (harm 2));can carry HARM munition
     (object fa-18 (service USN)
            (radius 550)
            (speed 420)
            (air-self-protect 7)
            (harm 2) ;From Murray p92; can maybe carry more!!
            (intercept 7)) ;Equiv. to F-15E in capability
     )
(class bomber
     (object b-52
            (burn-rate 20000)
            (radius 7650)
            (speed 360)
            (cas 6))
     (object b-1b
            (burn-rate 14000)
            (radius 3500)
            (speed 540))
     (object b-2 (stealth 1)
            (burn-rate 14000)
            (radius 2600)
            (speed 420))
     )
```

```
(class sead;other AC have SEAD capab.
     (object ea-6b (service USN)
            (harm 4);!4 or 2??
            (jam-comm 180)
            (jam-radar 180)
            (radius 695)
            (speed 420))
     (object ec-130
            (jam-comm 180)
            (radius 1300) ;!?
            (speed 240))
     (object ef-111
            (jam-comm 180)
            (jam-radar 180)
            (radius 3100)
            (speed 540))
     (object f-4g
            (harm 4);!do some only carry 2?
            (radius 680)
            (speed 480))
     )
(class tanker
     ;;max-offload takes into account reserves, takeoff cost, etc.
     (booms 1)     ;!some kc-10s can have 2
     (object kc-135;assume the KC135R model
            (burn-rate 10000)
            (radius 5000) ;!just a guess
            (speed 440)
            (max-offload 145000))
     (object kc-10
            (burn-rate 20000)
            (radius 5000) ;!just a guess
            (speed 520)
            (max-offload 272000))
     )
(class reconaissance
       (optical 1)        ;assume all recon aircraft have cameras
       (radar 1)          ;!added for MPA 9/96 demo
       (object rf-4);!may not be in IFD-4 scenario
       (object u-2))
(class awacs
  (object e-3
       (max-sortie-duration 8))
  (object e-2 (service USN)
       (max-sortie-duration 6))
     )
(class ground-attack;close-air support
     ;; Any strike AC can do CAS; these are specialized for CAS
     (cas 9)              ;prefer these aircraft for CAS roles
     (object a-10a
            (radius 250)
            (speed 260))
```

```
          (object ac-130
                 (radius 1300)
                 (speed 240))
          (object av-8b (service USMC)
                 (radius 595)
                 (speed 260)
                 (air-self-protect 2));Just a guess
          )
       (object jstars);supports CAS - can see vehicles
       (object abccc);supports CAS -C^2
       ))
```

5.9 TARGET-TACTIC CLASS

The target-tactic class enumerates the variety of ways in which a target can be attacked, besides delivering munitions to it. Limited use of it is made in the ACP KB; it is intended as a hook for future expansion.

```
(class target-tactic
       ;;Various ways besides STRIKEs that a target may be attacked.
       ;;Used in VULNERABLE-TO predicates
       (object sof-attack) ;Special operations
       (object jam-attack) ;Of comm, radar. !Not used in TIE97
       (object helo-attack) ;Attack helicopters. !Not used in TIE97
       )
```

5.10 CAPABILITY CLASS

The capability class enumerates the various enemy capabilities that are modeled.

```
(class capability                ;ie. of a target network
       (object test-capab)       ;for testing only
       (object c3)
       (object higher-command)   ;higher HQ/cmd
       (object aaa-engagement)   ;ability to fire antiaircraft artillery
       (object sam-engagement)   ;ability to launch and guide SAMs
       (object air-operations)   ;airbase’s ability to conduct air operations
       (object air-picture)      ;radars etc. that allow reacting to attacks
       (object air-intercept)    ;engaging attacking blue aircraft with ftrs
       (object sector-intercept) ;defense of a sector with fighters
       (object air-attack)     ;attacking in-air blue aircraft with ftrs
       (object air-strike) ;attacking blue territory or ships with aircraft
       (class target-capability
       ;;These are capabilities that, when paired with a TARGET in a
       ;;network definition, can be degraded by attacking that target.
          (object c3-air)
          (object early-warning)  ;ie. of hostile aircraft
          (object ground-control) ie. of friendly aircraft
       (class airbase-capability
            (object takeoff-landing ;ie. runways and taxiways
            (object aircrew)
```

```
          (object aircraft)     ;aircraft used to shoot down blue aircraft
          (object fuel)
          (object munitions)
          (object maintenance))
     (object sam-launcher)      ;launchers+missiles at SAM site
     (object sam-support)       ;nonweapon req’ts for SAM to function well
     (object electricity)
     (object communications)
 ))
```

5.11 COMPOSITION CLASS

The composition class enumerates the three ways in which the effectiveness of a parent network depends on the effectivenesses of its components.

```
(class composition
       (object additive) ;Degradation is proportional to that of components
       (object critical) ;Degrading any one component degrades the entire net
       (object redundant);All components must be degraded
)
```

5.12 THREAT CLASS

The threat class enumerates the various threats that are modeled. It is used primarily in THREAT predicates to quantify threats and associate each with a target network.

```
(class threat
       (object aaa-threat) ;to blue aircraft over red airspace
       (class threat-aircraft
       (object oca-threat) ;to blue aircraft over blue territory
       (object intercept-threat);to blue aircraft over red airspace
       (object strike-threat) ;to blue ground-based forces
       (object sea-strike-threat) ;to blue sea-based forces
       )
(class threat-missile
       (object sam-threat) ;to blue aircraft
       (object tbm-threat) ;to blue ground-based forces
       ))
```

5.13 E-AIRBASE CLASS

The e-airbase class is used to enumerate enemy airbases. These are used in THREAT predicates, and in predicates defining target networks. Objects of this class are typically generated programmatically from a data file of targets.

```
OBJECT: Frans-Kaisiepo-Airbase
 PARENT-CLASS: E-AIRBASE
 PROPERTIES:
  SECTOR = FRANS-KAISIEPO-SECTOR,,
  COUNT = 1,
```

```
  LAT = 112,
  LON = 13607,
  LATBOX = 100,
  LONBOX = 13600;
END OBJECT
```

5.14 AIRBASE CLASS

The airbase class is used to enumerate friendly airbases.

```
;;; ALL BLUE BASE NAMES MUST BE <= 8 CHARACTERS due to CTEM limitations
;;;! Note all airbases also have a LOCATED-WITHIN predicate
(class airbase            ;ie. friendly
       (lat 9000) (lon 0)         ;Make sure every base has a lat/lon
       (object NO-AIRBASE)        ;for assets not at any base (sensors?)
       (class carrier ;Assume fixed loc. for duration of plan
              (object nimitz (lat 1500) (lon 14300))
              (object kennedy (lat 0000) (lon 14300))
       (class air-field  ;An airbase on land
              (object port-moresby (lat 0926) (lon 14713))
              (object wewak (lat 0335) (lon 14340))
              (object nadzab (lat 0634) (lon 14644))
              (object darwin (lat 1200) (lon 13100))
              (object cairns (lat 1700) (lon 14530))
              (object guam (lat -1328) (lon 14447))
              (object okinawa (lat -2700) (lon 13400))
              (object whiteman (lat -4730) (lon -11100))
              (object hickam (lat 2115) (lon 15756))) ;Hawaii
       )
```

5.15 SEAPORT CLASS

The seaport class can be mentioned in COG predicates to identify a friendly center of gravity.

```
(class seaport  ;;These are COGs that must be protected
       (lat 9000) (lon 0)               ;Make sure every base has a lat/lon

       (object port-moresby-seaport (lat 0926) (lon 14713))

       )
```

5.16 MISSION-CATEGORY CLASS

The mission-category class is used in post-CTEM planning to categorize support mission resource requirements.

```
(class mission-category
       ;;All missions (primitive actions) in plan fall into one of these groups
       (object strike)
       (object counterair);! was called ESCORT for IFD4
```

```
        (object isr)
        (object ew)
        (object refuel)   ;! was called FUEL for IFD4
        )
```

6 OPERATORS, GOALS, AND ADVICE

The ACP KB solves two separate planning problems—the pre-CTEM plan and the post-CTEM plan. Disjoint sets of operators are used for each plan. Each set is discussed separately.

The ACP KB includes a set of operators for refining goals into actions and subgoals. Operators fall naturally into groups based on the types of goals that each refines. These groups are termed *abstraction levels*. Operators within an abstraction level typically refine the same goal or related goals. The following discussion of operators is organized by abstraction levels.

6.1 PRE-CTEM PLANNING

The pre-CTEM portion of the KB contains the following abstraction levels:

- High-level decisions
- Defensive/offensive substrategies
- Effectiveness reduction (of networks)
- Target selection
- Attack selection
- Effectiveness computation.

The source file `airsup-ops.sipe` contains the operators for the first two abstraction levels. The remainder are contained in `ops.sipe`, except that effectiveness computation is done in `ops.sipe` and `side-eff-ops.sipe`, and also by deductive rules (in `deduce.sipe`).

6.1.1 HIGH-LEVEL DECISIONS

The highest-level goal in pre-CTEM planning is simply to achieve air superiority in the area of interest in the scenario. The ACP KB first partitions the air superiority problem into its offensive and defensive components. The offensive and defensive air superiority subgoals are solved separately, though there can be interactions between them (e.g., neutralizing airbases can serve both goals).

The ACP KB contains two options for the degree of air superiority desired: Air Superiority and Air Supremacy. Air Supremacy forces threats and air defenses to be reduced to lower levels than Air Superiority; otherwise, these subgoals are handled identically.

The operators implementing these alternatives are achieve-offensive-air-superiority, achieve-offensive-air-supremacy, achieve-defensive-air-superiority, and achieve-defensive-air-supremacy. The defensive and offensive alternatives are described in the following two subsections (Subsections 6.1.2 and 6.1.3).

6.1.2 DEFENSIVE AIR SUPERIORITY SUBSTRATEGIES

SIPE-2 plans for defensive air superiority by responding to all threats to all friendly COGs that are present in the scenario. A THREAT-AXIS predicate is used to associate threats with the places each threatens, and where the threat would cross the border:

```
(THREAT-AXIS <threat> <place> <threatened-place> <border-sector>).
```

Each threatened-place that is, or contains, a friendly COG causes the ACP KB to plan some response to the threat. The response could be preemptive (i.e., the threat is attacked somehow) or defensive in nature. If the threat is deemed insignificant, the ACP KB notes this with an IGNORE-THREAT action.

The following sequence of decisions is made by SIPE-2 in planning defensive air superiority:

1. Either a PROTECT or PROTECT+ROLLBACK overall strategy is adopted. Both will protect all threatened COGs somehow; in addition the latter will preempt threats that are close to friendly territory. Only one rollback strategy is implemented completely in the ACP KB—that of preempting all threats originating in sectors bordering friendly territory. Other strategies, perhaps based on strength or type of threat or proximity to COGs, could be implemented by writing additional operators.
2. All significant threats to all blue COGs are identified. This is a deterministic step in which all threat-COG combinations are found, and protection goals are posted for each.
3. A protection alternative is chosen from the following:
    - A. Combat air patrol (CAP) at the threatened place
    - B. CAP in the sector at which the threat axis intersects the border
    - C. CAP over the origin of the threat
    - D. Preempt the threat (i.e., reduction of the threat to an “acceptable” level).
4. Support missions are generated.

6.1.3 OFFENSIVE AIR SUPERIORITY SUBSTRATEGIES

Planning for offensive air superiority uses a “breach and extend” strategy. A breach is a point at which the IADS is initially degraded enough to permit ingress by nonstealth aircraft. Once a breach is achieved, air superiority is extended over areas that are to be attacked as part of the overall air campaign. Both the breach location(s) and the areas to which air superiority is to be extended are expressed in sectors.

Air superiority in a sector is attained by reducing the SAM-THREAT and INTERCEPT-THREAT in that sector to acceptable levels. SAM-THREATs typically include all SAMs in that sector, even if some SAMs cover only a small part of the sector. INTERCEPT-THREATs typically include all fighters whose range permits operation over the sector.

The ACP KB implements the following breach alternatives:

- One sector
- Two sectors (possibly widely separated or on different borders)

- All sectors along an entire hostile border.

Furthermore, the ACP KB implements alternatives for restricting the breach sector(s) to inland sectors, coastal sectors, or sea sectors.

Currently, the sectors to which air superiority is to be extended are explicitly specified as part of the scenario. These sectors are those in which air strikes are called for in the overall plan. These sectors are designated by a REDUCE-IADS-THREAT predicate.

In principle, these sectors could be inferred from the rest of the air campaign plan by simply noting the sectors in which attacks are to be conducted. Because, however, the KB considers only the air superiority portion of the plan, we decided to leave the implementation of such an inference technique for future work.

6.1.4 EFFECTIVENESS REDUCTION

The third abstraction level in the ACP KB reduces the effectiveness of target networks. This abstraction level contains three sublevels—threat reduction, network reduction, and network degradation.

6.1.4.1 Threat Reduction

The higher abstraction levels of the ACP KB select threats whose levels are to be reduced. This selection is done during defensive air superiority planning by preempting threats such as STRIKE-THREATs, and during offensive air superiority by reducing INTERCEPT-THREATs and SAM-THREATs over enemy territory. Since each threat is associated with one network, this process leads to a deterministic selection of networks whose effectivenesses are to be reduced.

6.1.4.2 Network Reduction

The operators at the network reduction level provide a general-purpose capability to reduce the effectiveness of a given network so that it falls below a given threshold level. In this version of the ACP KB, only networks associated with threats are reduced. However, the operators are suitable for reducing any kind of network.

Each operator has as its purpose a LEVEL<= predicate. Expansions of the operator generate actions that reduce the effectiveness of the network below the threshold level. The reduction is accomplished by posting a network degradation goal for one or more of the components of the network being reduced. When the network degradation goal is fully expanded, the LEVEL<= predicate is checked, and is solved again recursively if the effectiveness is not below the desired threshold.

The ACP KB implements three network reduction alternatives: reduce-network-serial, reduce-network-parallel, and reduce-network-completely. The first selects only one component to be degraded. In the second, all components are degraded to some extent. In the third, all components are attacked, as are all components of components, until all targets supporting the network are attacked.

6.1.4.3 Network Degradation

The operators at the network degradation level have as their purpose a DEGRADE-CAPABILITY (or a DEVASTATE-CAPABILITY) predicate, which specifies a network whose effectiveness is to be reduced by an arbitrary amount. This differs from the purpose of the network reduction operators, which specify an explicit level to which the effectiveness is to be reduced.

Degradation is accomplished by expanding the operator into one or more degradation goals for components of the network being degraded. These are expanded recursively until components that are targets (as opposed to networks of targets) are reached. At that point the recursion terminates, and a technique for attacking the target is selected. Devastation (as opposed to degradation) means that all components of a network are attacked recursively, effectively attacking all targets supporting a network.

The ACP KB implements numerous network degradation alternatives:

- The degrade-best-network-component selects the one component with the highest contributing weight for degradation.
- The degrade-one-network-component selects one arbitrary component for degradation.
- The degrade-all-network-component selects all components for degradation.
- degrade-network-completely component devastates all components.

The above operators apply to any kind of network. In addition to these, the ACP KB can encode special-purpose degradation strategies for specific kinds of networks. An example is the blind-air-picture operator, which degrades the GROUND-CONTROL and EARLY-WARNING components of a network that provide an AIR-PICTURE of a sector.

6.1.5 Target/Attack Selection

The target/attack selection abstraction level contains two sublevels—target selection and attack selection.

Target selection terminates the recursion started by the network degradation operators. Attack selection terminates pre-CTEM planning by generating primitive actions to attack selected targets.

6.1.5.1 Target Selection

The target selection sublevel consists of an auxiliary set of operators that serve to terminate the recursion started during network degradation or devastation abstraction levels. These operators are degrade-target-component and devastate-target-component. They trigger the computation of the change in effectiveness of the networks containing the target by posting an UPDATE-PARENT effect. They also post a TARGET goal to be expanded by the attack selection operators.

6.1.5.2 Attack Selection

The attack selection operators have as their purpose a TARGET goal, and expand into a primitive action that is a kind of attack on the given target. Two alternatives are implemented—make-strike-target and make-sof-target. The former is an attack by delivery of munitions. The latter is an attack by special forces, the details of which are unspecified. Targets must be identified individually as being vulnerable to a nonmunition attack via a VULNERABLE-TO predicate. Other attacks, such as jamming, are not implemented, but are natural extensions to the ACP KB's capabilities.

6.1.6 Effectiveness Computation

The effectiveness computation operators are used not to reflect planning decisions, but rather to compute the changes of effectiveness in networks, given changes to their components. The general technique used is to defer the computation of the effectiveness of a parent network during recursive degradation and reduction until its children's effectivenesses are computed.

This is done by posting a RESOLVE-NETWORK-DEGRADE (or -REDUCE) goal serially after a DEGRADE-CAPABILITY goal. The RESOLVE goal is copied down each planning level by the appropriate DEFER operator, until a change occurs in the network being degraded. When such a change occurs, the appropriate RESOLVE operator computes the new effectiveness of the network, and posts a change to it. This process proceeds recursively until all attacks on networks are resolved.

A change in a component is detected by the presence of one of the following predicates: MAXLEVEL (for REDUNDANT networks), MINLEVEL (for CRITICAL networks), or CONSUME (for ADDITIVE networks). These predicates are posted as effects by the deductive rules that participate in the computation of effectivenesses.

The operators for resolving degrades are defer-redundant-degrade, resolve-redundant-degrade, defer-critical-degrade, resolve-critical-degrade, defer-additive-degrade, and reserve-additive-degrade.

The operators for resolving reductions are resolve-critical-reduce, defer-critical-reduce, defer-redundant-reduce, resolve-redundant-reduce, and resolve-additive-reduce.

The ACP KB propagates changes in effects upwards to all parent networks of an attacked component. It does so even if the parent itself did not occur in a goal to degrade or reduce the component; such changes are termed *side effects* or *collateral effects*.

The ACP KB permits the specification of advice, which disables the computation of collateral effects. This can improve performance and result in the generation of smaller plans containing fewer levels.

The operators that propagate effectiveness changes upwards are resolve-attack-side-effects, resolve-root-attack, ignore-parent-child-side-effects, resolve-parent-child-main-effect, resolve-parent-child-side-effects, resolve-additive-effect, resolve-critical-effect, and resolve-redundant-effect.

6.1.7 Advice

The ACP KB incorporates advice as specified in Myers [1997]. It uses *features* both to characterize distinctions between alternative operators, and to specify a context in which pieces of advice are active. It uses *roles* to specify planning arguments whose values are constrained by advice.

The ACP KB implements the following features:

- `DEFENSIVE DEFENSIVE-AIR-SUPERIORITY DEGRADATION ECONOMICAL`

- `ESTABLISH-INGRESS IADS-REDUCTION INFLICT-BLINDNESS`
- `INTERCEPT-REDUCTION MASS OFFENSIVE OFFENSIVE-AIR-SUPERIORITY`
- `REFUELING SAFE SAM-REDUCTION THREAT-DEFENSE THREAT-REDUCTION`.

The ACP KB implements the following roles:

- `COG-CAPABILITY COG-PLACE THREAT-TYPE`
- `PARENT-CAPABILITY GRANDPARENT-CAPABILITY INGRESS-PLACE`.

Using these features and roles, the ACP KB implements numerous pieces of advice. Each piece of advice has a one-sentence description associated with it, to identify it in the user interface for advice selection. Following are the descriptions for all implemented pieces of advice:

- Do not compute the side effects of attacks on target networks
- Choose SENTANI-SECTOR as a point at which to breach the IADS
- Choose SCHANJOK-SECTOR as a point at which to breach the IADS
- Choose AGATS-SECTOR as a point at which to breach the IADS
- Choose TUSCON-SECTOR as a point at which to breach the IADS
- Choose SORONGWATI-SECTOR as a point at which to breach the IADS
- Choose NOJIMSAN-SECTORS as a point at which to breach the IADS
- Choose SARMI-SECTOR as a point at which to breach the IADS
- Choose ROCKATOON-SECTOR as a second point at which to breach the IADS
- Don’t consider AGATS-SECTOR or SCHANJOK-SECTOR as INGRESS points
- Neutralize enemy intercept capability by denying them their air picture
- Blind enemy radars in order to deny them their air picture
- Neutralize enemy air C3 in order to deny them their air picture
- Disable SAMs by attacking launchers directly
- When attacking SAM launchers, attack ALL launchers
- Use non-preemptive operations when achieving defensive air superiority
- Use F-14s to man all point-defense CAPs over carriers
- Use F-15Cs to man all point-defense CAPs over airfields
- Preempt all threats to naval surface ships
- Preempt air operations by attacking airbase munitions
- Choose runways/taxiways as targets when preempting strike threats
- When attacking airbases, mass forces against one target type
- Use non-preemptive operations when achieving defensive air superiority
- Breach IADS at two different sectors
- Achieve defensive air supremacy
- When defending against threats, use BARCAPs.

6.2 POST-CTEM PLANNING

Post-CTEM planning takes as input a set of SUPPORT-PKG goals. Each such goal designates the type and number of strike aircraft participating in an attack on a set of targets. The ACP KB expands each of these goals into primitive actions for the strike mission itself, and for any support missions required by the strikers. These missions include escort/counterair (protection from enemy aircraft); suppression of enemy air defense (SEAD), i.e., protection from enemy SAMs; refueling; and reconnaissance (both prestrike and poststrike).

To do a high-fidelity estimate of required support, the flight profile (the exact route from base to refueling to target, the altitude, etc.) needs to be considered. The ACP KB does not generate or represent the flight profile of strikes; it assumes a typical altitude profile and straight-line paths between bases, refueling points, and targets. Furthermore, it assumes that all targets in a package are colocated; in reality, targets within a package can be separated by as much as 100–200 miles. Also, in determining requirements for protective support, some important factors like the criticality of the target being attacked and the value of the strike aircraft are not considered.

The operators for post-CTEM planning are in the source file `strike-ops.sipe`. The abstraction levels used in post-CTEM planning are as follows:

- Decompose by striker type
- Select SEAD protection
- Select air protection
- Add reconnaissance
- Add tankers and output support missions.

Each of these levels is described in detail in the remainder of this section.

There is a correspondence between the abstraction levels and the kinds of support missions that are planned with the ACP KB: SEAD, air protection, and reconnaissance missions are added in successive abstraction levels. In addition, the last level adds tanker missions to fulfill package refueling requirements.

6.2.1 Decompose by Striker Type

Each SUPPORT-PKG goal is classified as one of three types, based on the kind of striker: unmanned, stealth, or regular. Each has different support requirements. An unmanned (e.g., cruise missile) strike requires only reconnaissance support. Stealth aircraft, by their nature, are safer when flying alone; they require only reconnaissance and possibly refueling support. Regular (nonstealth) packages may require SEAD and air protection, in addition to reconnaissance and refueling.

The operators are support-nonstealth-pkg, support-cruise-missile-strike, and support-stealth-strike.

6.2.2 Select SEAD Protection

These operators add SEAD protection as needed, based on the SAM-THREAT in the sector that is being attacked. SEAD support consists of aircraft that are capable of firing high-speed antiradiation missiles (HARMs) at SAM radar sites. This type of SEAD is termed *lethal SEAD*.

Lethal SEAD is usually supplemented with a capability to jam SAM radars; this is termed *nonlethal SEAD*. Hooks are present in the ACP KB to allocate nonlethal SEAD; however, this kind of support is unimplemented.

Two options are implemented: dedicated SEAD support (the operator add-dedicated-sead), and self-protection (the skip-sead-no-threat). The former is used in the case of a high-threat environment, the later in a low-threat environment. A third option of interest, protection via a SEAD CAP, is not implemented. It is a middle course between self-protection and dedicated SEAD support, allowing SEAD support to be shared among packages flying in a certain area.

6.2.3 Select Air Protection

These operators add protection from attack by enemy aircraft. Support consists of aircraft with the capability of shooting down enemy interceptors.

Four options are implemented: no protection, self-protection, fighter sweep, and dedicated escort. No protection is applicable in an environment where the enemy INTERCEPT-THREAT is negligible. Self-protection is applicable in a low-threat environment in which the strike aircraft are at least somewhat capable in air-to-air combat. Fighter sweep is applicable in medium-threat environments and lower. Dedicated escort is applicable in all situations.

The number of aircraft used in a fighter sweep or dedicated escort is a function of the level of the threat and the size of the package (number of strikers plus number of dedicated SEAD aircraft). It is computed by the ACP KB function ESCORTS-REQUIRED. The number varies from two aircraft to roughly the number of aircraft being protected.

The air protection operators are skip-air-protection-for-unmanned, add-self-protection-air, add-fighter-sweep, and add-dedicated-escort.

6.2.4 Add Reconnaissance

The ACP KB contains a notional capability for estimating the reconnaissance support required by strikes. Due to a variety of factors, including security classification issues regarding the nature of reconnaissance and the inherent sensitivity of these capabilities, only a notional capability was implemented.

Reconnaissance support consists of adding a PRESTRIKE-RECON action that precedes the attack by several days (in order to allow time for the data to be gathered, analyzed, and distributed), and a bomb damage assessment (BDA) action that follows the attack. Two options were implemented: reconnaissance on a per-target basis, and on a per-package basis. The former generates a very large number of actions, but reflects the scope of the problem of integrating reconnaissance with operational planning. The latter is used in practice as it generates smaller plans; however, it is unrealistic to assume that one reconnaissance action will suffice for all targets within a package. In reality, several sensors of different kinds may be required.

The operators are add-support-with-pkg-recon, add-unmanned-support-with-pkg-recon, and add-support-with-target-recon.

These operators also perform the auxiliary step of decomposing the support goal into its components of strike, escort/air protection, SEAD, and refueling. The PRESTRIKE-RECON and BDA actions are sequenced before and after these components, respectively.

6.2.5 Add Tankers and Output Support Missions

These operators take the decomposed support goals posted in the above level, and expand each into the appropriate primitive action. Except for the selection of tanker type for refueling, this step is deterministic, as all information needed to characterize the action has been previously determined.

It is at this level that tanker requirements for refueling are computed for each package, by the ACP KB function PKG-REFUELERS. This function takes into account the type of tanker, the location of the targets and the refueling point, and the type and number of all aircraft in the package.

Refueling points are specified by the predicates

`(REFUEL-FOR-SECTOR sector lat lon)`

`(REFUEL-FOR-BLUE lat lon).`

The first predicate allows the specification of refueling points for individual sectors being attacked. The second allows the specification of catch-all refueling points.

The operators in this level and the associated primitive actions generated are shown in Table 1.

Table 1. Add Tankers and Output Support Missions

OPERATOR	ACTION
make-strike	strike
make-bda	bda
no-tanker-orbits-needed-for-unmanned	no-tankers-needed
make-tanker-orbits	tanker-orbits
make-fighter-sweep	fighter-sweep
no-lsead-needed	no-lethal-sead-needed
make-lsead	lethal-sead
no-escorts-needed	no-escort-needed
make-escorts	escort

These operators select a preferred type of asset (aircraft) to perform the primitive action. In numerous demonstrations, the actions output by the KB are postprocessed to identify and repair shortfalls in assets. In order to identify alternative asset allocations for such postprocessing, each primitive action has a set of ASSET-USAGE predicates, which specify allowable types and numbers of assets suitable for performing the action.

7 DEDUCTIVE RULES

Several simple deductive rules in the ACP KB compute simple relationships among objects (for example, the commutativity of ADJACENT relationships among geographic sectors).

Other deductive rules localize the computations needed to compute network effectivenesses and associated threat levels. These update rules, working in conjunction with operators in the Effectiveness Computation abstraction level, compute and propagate changes to network effectivenesses.

The update rules are update-threat-level, update-additive-effectiveness, update-critical-effectiveness, and update-redundant-effectiveness. The first of these rules is triggered nonrecursively whenever the network associated with a threat changes. It computes and posts the proportional change to the threat level.

The remaining update rules work as follows. Each is triggered by an UPDATE-PARENT predicate for a network of the associated type (additive, critical, or redundant). Each computes and posts an effect to reflect the change in the parent network caused by the change to the child. For additive networks, this is a CONSUME effect. For critical networks, this is a MINLEVEL effect. For redundant networks, this is a MAXLEVEL effect. The effect causes a RESOLVE operator, rather than a DEFER operator, to be applied, which in turn triggers another UPDATE-PARENT effect for the parent. This process repeats recursively until changes to all top-level networks (and associated threats) are resolved.

8 PREDICATES

Predicates are used to model the static world state that is defined by the scenario. They are also used to model the dynamic world state that is changed due to actions planned by SIPE-2.

The predicates described here are used to capture the intelligence analysis that is part of the scenario. This analysis includes a target network specification and threat characterization.

8.1 TARGET NETWORKS

A network is designated with a NET predicate:

`(NET <capability> <place> <composition>)`

e.g.,

`(NET MUNITIONS Sentani-Airbase ADDITIVE).`

Each network has a capability and a place associated with it. The place specifies the extent or the coverage of the capability. It can be a point location, a sector, or a region. Each network is one of three types, or *compositions*: ADDITIVE, CRITICAL, or REDUNDANT. The composition of a network determines how the effectiveness of its parent network is computed as a function of its components. ADDITIVE means that the parent's effectiveness is a weighted sum of its components. CRITICAL means that the parent's effectiveness is the minimum effectiveness of all its components. REDUNDANT means that the parent's effectiveness is the maximum effectiveness of all its components.

In addition, all connections between a component and all its parent must be specified via a PROVIDES predicate:

```
(PROVIDES <parent-net> <child-net> <weight>)
```

e.g.,

```
(PROVIDES AIR-OPERATIONS Sentani-Airbase MUNITIONS LAX 1.0)
(PROVIDES MUNITIONS Sentani-Airbase MUNITIONS AMMO-BUNKER-A 0.6)
(PROVIDES MUNITIONS Sentani-Airbase MUNITIONS AMMO-BUNKER-B 0.4).
```

A weight reflecting the importance of that component must be provided. Modeling a sophisticated IADS, complete with C^3, power generation, and EW/GCI* coverage for each connection can involve many hundreds or even thousands of connections.

Intrinsic capabilities are designated with NEEDS predicates:

```
(NEEDS <net> <capability> <min-effectiveness>)
```

e.g.,

```
(NEEDS COMMUNICATIONS NOJIMSAN-SECTOR ELECTRICITY 0.4).
```

< min-effectiveness> is a value between 0 and 1; it is the effectiveness level to which the target network can be reduced by being denied all external sources of <capability>. In the example, a COMM network requires ELECTRICITY to function, but has backup generators that enable it to operate at 40% efficiency if external power is cut.

Each network has an effectiveness between 0 (inoperative) and 1 (fully capable), implemented as a LEVEL predicate:

```
(LEVEL <capability> <place> <when> <level>)
```

where <level> reflects the effectiveness of the network <capability> <place> on the day designated by <when>. The INITLEVEL predicate is used to initialize a network level for all days of the scenario:

```
(INITLEVEL <capability> <place> <level>).
```

Here is an example network which models the air intercept network in the western region of East Cyberland as a function of the airbases in or near that region. The weights reflect an intelligence estimate of the degree of intercept threat emanating from the airbase, which is based on the number and type of intercept aircraft at the base:

```
;;Enumerate all airbases with intercept threats, weighted by threat strength

(net air-intercept EASTERN-WC ADDITIVE)
(provides air-intercept EASTERN-WC air-operations Sentani-Airbase .45)
(provides air-intercept EASTERN-WC air-operations Schanjok-Airbase .20)
(provides air-intercept EASTERN-WC air-operations Nojimsan-Airbase .35)
```

8.2 THREATS

A threat is designated by a THREAT predicate:

```
(THREAT <threat> <net> <daynight> <initial-strength>)
```

e.g.,

---

*C^3: command, control, and communications; EW/GCI: early warning/ground controlled interception.

```
(THREAT SAM-THREAT SAM-ENGAGEMENT ROCKATOON-SECTOR 24-HOUR 150).
```

Each threat is associated with one network, and is further characterized as daylight-only, nighttime-only, or 24-hour threat.

A threat is reduced by attacking the underlying network; its current strength is represented by a LEVEL predicate:

```
(LEVEL <threat> <net> <daynight> <day> <strength>).
```

A threat is related to a place it threatens by a THREAT-AXIS predicate:

```
(THREAT-AXIS <threat> <origin> <threatened-place> <border-sector>)
```

e.g.,

```
(THREAT-AXIS strike-threat Sentani-Airbase wewak port-bobozel-sector).
```

The <border-sector> is the place where the threat crosses the border en route to <threatened-place>. If multiple crossings are possible, one THREAT-AXIS for each is specified.

Numeric threat levels are mapped to a symbolic threat rating by means of THREAT-RATING-LEVEL predicates, as shown in Table 2.

Table 2. Threat Rating Levels

RATING	LOW	HIGH
NONE	0	1
VERY-LOW	1	10
LOW	10	40
MED	40	70
HIGH	70	100
VERY-HIGH	100	9999999

Ratings are used in various threat reduction goals to specify symbolically the level of threat tolerable for air superiority. These predicates are used to determine the numerical reduction in the threat level that must be attained to satisfy the threat reduction goal. The <low> and <high> values are somewhat arbitrary; they are modeling parameters used in conjunction with threat levels to represent threats of differing strengths.

9 ACKNOWLEDGMENTS

This research was supported by Contract F30602-97-C-0067 with the Defense Advanced Research Projects Agency. It has been built upon research done under contracts F30602-95-C-0235 and F30602-95-C-0175.

10 REFERENCES

Bienkowski, M. 1997. *SOCAP-ACPT Technology Integration and Application*, ITAD-7112-AR-97-095, SRI International, Menlo Park, California (June).

Deptula, COL D.A. 1996. (pamphlet) *Firing for Effect: Change in the Nature of Warfare*, Aerospace Education Foundation, Arlington, Virginia.

Lee, Thomas J. and David E. Wilkins. 1996. Using SIPE-2 to integrate planning for military air campaigns, in *Trends and Controversies*, IEEE Expert, Decemper 1996.

Murray, W. 1995. *Air War in the Persian Gulf*, The Nautical & Aviation Publishing Company of America, Baltimore, Maryland.

Myers, K.L. 1997. *User Guide for the Advisable Planner, Version 1.1,* SRI International, Menlo Park, California.

Myers, K.L. 1998. *Towards a Framework for Continuous Plannnng and Execution,* in Proceedings of the AAAI Fall Symposium on Distributed Continual Planning.

Wilkins, D.E. 1999. *Using the SIPE-2 Planning System: A Manual for SIPE-2, Version 5.2*. SRI International, Menlo Park, California.