Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Ontological models for military courses of action

A.-C. Boury-Brisset
DRDC Valcartier

C. Champagne
DMR Consulting

Defence R&D Canada – Valcartier
Technical Memorandum
DRDC Valcartier TM 2003-320
September 2006

Canada

Ontological models for military courses of action

A.-C. Boury-Brisset
DRDC Valcartier

C. Champagne
DMR Consulting

Defence R&D Canada – Valcartier

Technical Memorandum
DRDC Valcartier TM 2003-320
September 2006

Author

Anne-Claire Boury-Brisset

Approved by

Éloi Bossé

Section Head – Decision Support Systems

Approved for release by

Gilles Bérubé

Chief Scientist

Abstract

The mandate of the Technology Investment Fund Project entitled *CoA Critiquing System for the Improvement of the Military Estimate Process* is to investigate new concepts for building intelligent systems dedicated to the analysis and critiquing of military courses of action. Providing commanders with advanced decision aids requires thorough understanding of the processes involved, their information requirements, and the development of formal domain models (ontologies) upon which reasoning processes can be based (e.g., knowledge-based systems, intelligent agents). The document presents planning and courses of action models reported in the literature. For each model we identify the constituent elements and give relevant aspects of ontology development, such as knowledge representation techniques, methodology of development used, and exploitation of the ontology developed. The objective is to compile enough information about the different models to help us construct a COA ontology tailored to our requirements.

Résumé

Le projet de Fonds d'investissement technologique *Système de critique de suites d'actions en vue de l'amélioration du processus d'estimation militaire* vise à étudier de nouveaux concepts pour la conception de systèmes intelligents destinés à l'analyse et à la critique de suites d'actions militaires. Fournir des aides à la décision avancées requiert une bonne connaissance des processus sous-jacents, de leurs besoins en information, et le développement de modèles formels du domaine (ontologies) sur lesquels les processus de raisonnement peuvent s'appuyer (p. ex. systèmes à bases de connaissances, agents intelligents). Ce document présente un ensemble de modèles ontologiques relatifs à la planification et aux suites d'actions qui ont été reportés dans la littérature. Pour chacun, nous tentons de capturer les éléments contenus dans ces modèles et fournissons les aspects pertinents reliés au développement de l'ontologie, à la représentation des connaissances, à la méthodologie utilisée, ainsi qu'à l'exploitation qui en a été faite. L'objectif est de fournir suffisamment d'informations sur ces projets en vue de la construction d'une ontologie pour les suites d'actions répondant à nos besoins.

This page intentionally left blank.

Executive summary

For several years DRDC Valcartier has conducted research in the military planning domain with the objective of providing military planners with decision support systems and intelligent tools that facilitate the planning process. The mandate of the Technology Investment Fund Project entitled *CoA Critiquing System for the Improvement of the Military Estimate Process* is to investigate new concepts for building intelligent systems dedicated to the analysis and critiquing of military courses of action. Providing commanders with advanced decision aids requires thorough understanding of the processes involved, their information requirements, and the development of formal domain models (ontologies) upon which reasoning processes can be based (e.g., knowledge-based systems, intelligent agents). In building large and complex intelligent systems in knowledge-intensive domains, it is impractical to develop knowledge bases from scratch. Thus, knowledge engineering promotes the design of formal domain models or libraries of reusable components that include both ontologies and problem-solving methods. From a knowledge modelling perspective, ontologies provide domain models that represent a shared understanding of the domain and can be reused across a wide range of applications in that domain. The use of ontology has attracted much attention within the artificial intelligence community. Common representation of plans has long been a subject of interest. Efforts to model plans indicate the difficulties of representing real-world domains and the need for rich representations of planning knowledge. To date, most models proposed in the military planning domain draw from DARPA (Defense Advanced Research Projects Agency) programs, in particular the Darpa Rome Planning Initiative (ARPI), High Performance Knowledge Bases (HPKB), and Rapid Knowledge Formation (RKF) programs.

The document presents planning and courses of action models reported in the literature. For each model we identify the constituent elements and give relevant aspects of ontology development, such as knowledge representation techniques, methodology of development used, and exploitation of the ontology developed. The objective is to compile enough information about the different models to help us construct a COA ontology tailored to our requirements.

Boury-Brisset, A.-C., Champagne, C. 2006. Ontological models for military courses of action. DRDC Valcartier TM 2003-320. DRDC Valcartier.

Sommaire

Depuis plusieurs années, RDDC Valcartier a mené des travaux de recherche dans le domaine de la planification militaire en vue de fournir des aides à la décision et des outils intelligents afin d'aider les militaires durant le processus de planification. Le projet de Fonds d'investissement technologique *Système de critique de suites d'actions en vue de l'amélioration du processus d'estimation militaire* vise à étudier de nouveaux concepts pour la conception de systèmes intelligents destinés à l'analyse et à la critique de suites d'actions militaires. Fournir des aides à la décision avancées requiert une bonne connaissance des processus sous-jacents, de leurs besoins en information, et le développement de modèles formels du domaine (ontologies) sur lesquels les processus de raisonnement peuvent s'appuyer (p. ex. systèmes à bases de connaissances, agents intelligents). La construction de systèmes d'envergure dans des domaines complexes requérant de nombreuses connaissances du domaine rend difficile la construction de bases de connaissances. L'ingénierie des connaissances promeut la conception de modèles formels du domaine ou de librairies de composantes réutilisables qui incluent à la fois des ontologies du domaine et des méthodes de résolution de problèmes. Les ontologies constituent des modèles d'un domaine qui représentent une compréhension partagée de ce domaine et peuvent être réutilisées dans différentes applications. La construction et l'exploitation d'ontologies a connu un intérêt soutenu en intelligence artificielle. En particulier, la représentation de plans a donné lieu à de nombreux travaux. Les efforts pour modéliser les plans révèlent les difficultés à représenter des domaines reliés au monde réel et la nécessité de représentations suffisamment riches des connaissances relatives à la planification. La majorité des efforts dans ce domaine sont issus de programmes DARPA (Defence Advanced Research Projects Agency) reliés soit au domaine de la planification (p. ex. Darpa Rome Planning Initiative), soit à la construction de grandes bases de connaissances tels que High-Performance Knowledge Bases (HPKB), et Rapid Knowledge Formation (RKF).

Ce document présente un ensemble de modèles ontologiques relatifs à la planification et aux suites d'actions qui ont été reportés dans la littérature. Pour chacun, nous tentons de capturer les éléments contenus dans ces modèles, et fournissons les aspects pertinents reliés au développement de l'ontologie, à la représentation des connaissances, à la méthodologie utilisée, ainsi qu'à l'exploitation qui en a été faite. L'objectif est de fournir suffisamment d'informations sur ces projets en vue de la construction d'une ontologie pour les suites d'actions répondant à nos besoins.

Boury-Brisset, A.-C., Champagne, C. 2006. Ontological models for military courses of action. DRDC Valcartier TM 2003-320. RDDC Valcartier.

Table of contents

Abstract/Résumé i

Executive summary iii

Sommaire iv

Table of contents v

List of figures vii

1. Introduction 1

2. Military planning process and related concepts 4

3. Planning models 6
   - 3.1 History 6
   - 3.2 KRSL/KRSL-Plans 7
   - 3.3 CPR 8
     - 3.3.1 Ontology content 8
     - 3.3.2 Knowledge representation 9
     - 3.3.3 Ontology extension and use 10
   - 3.4 I-N-OVA 11
     - 3.4.1 Ontology content 11
     - 3.4.2 Use of the ontology 12
   - 3.5 SPAR 12
   - 3.6 JFACC 15
     - 3.6.1 Context 16
     - 3.6.2 Ontology content 16
       - 3.6.2.1 Ontology import 17
       - 3.6.2.2 Ontology structuring 18
     - 3.6.3 Knowledge representation 20
     - 3.6.4 Use of the ontology 20

4. The DISCIPLE project 21
   - 4.1 Introduction 21
   - 4.2 DISCIPLE-COA 22
     - 4.2.1 Knowledge representation 22
     - 4.2.2 Ontology content 23
     - 4.2.3 Ontology import/export 24

4.2.4 Development methodology ... 25
4.2.5 Ontology development tool ... 26
4.3 DISCIPLE-COG ... 26
4.3.1 Centre of gravity ... 27
4.3.2 Ontology development and content ... 27
4.3.3 Ontology Import / export ... 30
4.3.4 Ontology development and use ... 31

5. PLANET ... 32
5.1 Ontology content ... 32
5.2 COA example ... 34
5.3 Knowledge representation ... 35
5.4 Use of the ontology and extensions ... 35
5.5 PLANET-COA application ... 35

6. Discussion ... 37
6.1 Ontology content examined in previous studies ... 37
6.1.1 General planning concepts ... 37
6.1.2 Specific military concepts ... 38
6.2 Development methodology ... 38
6.3 Knowledge representation formalisms ... 39
6.4 Ontolology import, merging and alignment ... 39
6.5 Relations to upper ontologies ... 39
6.5.1 CYC ... 40
6.5.2 SUMO ... 40

7. Conclusion ... 41

References ... 42

List of symbols/abbreviations/acronyms/initialisms ... 45

Internal Distribution list ... 46

External Distribution list ... 47

List of figures

Figure 1. History of planning models 6
Figure 2. KRSL-Plans high-level concepts 7
Figure 3. CPR meta-model 10
Figure 4. SPAR model (initial version - 1997) 13
Figure 5. SPAR Model (reviewed version - 1999) 15
Figure 6. The JFACC ontology (top-level view) 17
Figure 7. Organization of the JFACC ontology into its constituent modules 18
Figure 8. Architecture of Disciple 22
Figure 9. Fragment of the military organization ontology 23
Figure 10. Fragment of the military task ontology 24
Figure 11. Ontology fragment imported from CYC for Disciple-COA 25
Figure 12. Ontology management tools in Disciple 26
Figure 13. High-level concepts of COG ontology 28
Figure 14. Military Forces for COG 29
Figure 15. Planet ontology 34

List of tables

Table 1. Ontology content 19

This page intentionally left blank.

1. Introduction

For several years DRDC Valcartier has conducted research in the military planning domain with the objective of providing military planners with decision support systems and intelligent tools that facilitate the planning process. The mandate of the Technology Investment Fund Project entitled CoA Critiquing System for the Improvement of the Military Estimate Process is to investigate new concepts for building intelligent systems dedicated to the analysis and critiquing of military courses of action. In that context, this report provides a synthesis of research work reported in the literature on building formal military models in the planning domain that constitute the basis for building intelligent planning systems.

Military planning is a very complex process that involves the analysis of multiple parameters and copious data. Moreover, planning in large-scale real-world domains underscores the problem of the scope and breadth of knowledge required. Military planners have to deal with a wide variety of tasks (traditional military operations as well as humanitarian imperatives). This accentuates the challenge of building planning aids for knowledge acquisition and modelling.

Providing commanders with advanced decision aids requires thorough understanding of the processes involved, their information requirements, and the development of formal domain models upon which reasoning processes can be based (e.g., knowledge-based systems, intelligent agents). In building large and complex intelligent systems in knowledge-intensive domains, it is impractical to develop knowledge bases from scratch. Thus, knowledge engineering promotes the design of formal domain models or libraries of reusable components that include both ontologies and problem-solving methods. From a knowledge modelling perspective, ontologies provide domain models that represent a shared understanding of the domain and can be reused across a wide range of applications in that domain. Consequently, the design of ontologies is considered as a pre-requisite to the development of large knowledge bases.

Artificial intelligence (AI) concepts and techniques have been recognized as invaluable aids in acquiring knowledge assets and modelling and representing those knowledge assets at the appropriate level of formalization, and in designing intelligent decision support systems or knowledge repositories based on such models. During the last few years, the use of ontology has attracted much attention within the artificial intelligence community [25] because of its ability to formalize domain concepts and to facilitate information exchange among application programs. According to AI researchers, an ontology is defined as an explicit representation of a conceptualization [14]. It allows one to represent, in a more or less formal way, concepts of a domain of concern and how they interrelate. An ontology is represented using attributes, properties, relations between concepts, and may include constraints and axioms. But in the literature on ontologies and knowledge management, ontologies range from controlled vocabularies to formal computational models [17]. The common thread in this continuum is providing at least a shared understanding of the main concepts of some domain facilitating communication between humans or application programs.

Most significant modelling efforts in the military domain have been devoted to planning. Common representation of plans has been a subject of interest for a considerable time. Efforts to model plans underscore the difficulties of representing real-world domains and the need for rich representations of planning knowledge. Some studies reported in the literature aim to provide general plan models (e.g. O-PLAN, SPAR, PLANET), whereas others focus on the representation of courses of action (COA), i.e., outlines of plans and the entities of the domain involved in their development, and analysis of a given situation context (e.g., ontologies from the Disciple-COA project and Disciple-COG, which extends the COA ontology to represent the military concept of centre of gravity). In the context of military data models, the NATO LC2IEDM (Land C2 Information Exchange Data Model) [1] is also of great value because it stores the core data needed to describe information to be exchanged between C2 systems with respect to the battlefield domain.

Most models that have been proposed in the military planning domain draw from DARPA (Defense Advanced Research Projects Agency) programs.

First, the ARPI (DARPA Rome Planning Initiative) was a program (1991–1999) whose mission was to develop, demonstrate and transition advanced technology for automatic and interactive planning, scheduling and decision-making to allow better and faster planning in complex, stressed situations. Within this program, several general models were proposed to represent plans, activities and/or processes in order to support knowledge sharing about plans.

Later, new DARPA programs, namely the High Performance Knowledge Bases (HPKB) and the Rapid Knowledge Formation (RKF) programs, focused on the building of large knowledge bases in support of military applications. Consequently, the participants in these programs contributed to advances in the field of knowledge representation technology in general, but also provided relevant models and tools dealing with course of action critiquing and analysis, which was a proposed challenge problem of the HKPB/RKF programs.

The High Performance Knowledge Bases (HPKB) DARPA Program (1997-1999) [11] was a research program to advance the technology of how computers acquire, represent and manipulate knowledge, with an emphasis on the development of very large knowledge bases by AI experts and computer scientists. The HPKB Program has demonstrated the usefulness of large knowledge bases and the feasibility of large-scale reuse. Different challenge problems have been proposed for building large knowledge bases, namely crisis management problem, workaround problem, and course of action analysis and critiquing. It has produced reusable knowledge libraries including an upper ontology and middle theories for crisis and battlefield reasoning. The Rapid Knowledge Formation (RKF) DARPA Program (1999-2003) [15] aimed at enabling distributed teams or subject matter experts (SMEs) to enter and modify knowledge directly and easily without the need for specialized training in knowledge representation, acquisition and manipulation. The RKF research program builds on HPKB contributions by enabling military and technical SMEs to directly encode their problem solving expertise and take over developing, extending and expanding the knowledge base to one million axioms.

The remainder of this report is organized as follows. In the next section, we give some definitions relevant to the military planning domain. Then, the main part of the document is composed of a synthesis of different projects dealing with planning and COA modelling for different purposes. For each of them, we introduce the context and objective of the work, we identify the constituent elements of the ontological models, and give relevant aspects of ontology development such as knowledge representation techniques, methodology of development, and exploitation of the ontology developed. This is followed by a synthesis of those projects with the aim of highlighting the contents of ontological models most suitable for planning, the knowledge representation techniques that are considered the most effective, and the issues affecting reusing, importing and merging ontologies. The objective is to compile enough information about the different projects to facilitate construction of a COA ontology tailored to our own purposes.

2. Military planning process and related concepts

In this section, we first provide the context of the work by describing the operational planning process that encapsulates course of action development, evaluation and ranking. Although the aim of this section is to describe the process and not the domain concepts, we try to highlight both the important concepts of the process and the relevant knowledge elements that are considered during the process.

The Canadian Forces Operational Planning Process (CFOPP) [18] is a coordinated process to determine the best method of accomplishing assigned tasks and to plan possible future tasks. It is comprised of several phases:

1. initiation;
2. orientation, to determine the nature of the problem and confirm the results to be achieved;
3. course of action development (the main part of the estimate process), resulting in a range of courses of action that can accomplish the mission;
4. course of action selection based on specific factors facilitating COA comparison;
5. plan development;
6. plan review.

In this context, a course of action is a possible option open to the commander that would accomplish the mission. It is an outline of a plan that is first described in broad terms and then refined with further details during analysis. A COA is not a complete plan in that it lacks many of the details necessary to execute the operation. But it provides a framework suitable for an appreciation of all factors and deductions to determine the viability of the various options. In particular, to be viable, each COA must meet the following essential criteria: suitability, feasibility, acceptability, and compliance.

The mission statement resulting from the orientation stage must answer the following statements [27].

- Who will execute the action? What type of forces?
- What type of action is contemplated (e.g., attack, defend)?
- When will the action begin?
- Where will the action take place (area of operations and objective)?
- Why (for what purpose) will each component of the force conduct its part of the operation?

During the estimating process, many domain entities are considered in the analysis of the mission and the development of COAs. Elements of the situation include environment, geography, political, enemy forces, friendly forces and logistics. Other important elements considered during the COA development process should be included in a COA ontology. These are:

- assumptions

- objectives
- restraints, constraints
- end state
- centre of gravity/decisive points
- lines of operation
- tasks (assigned, implied)
- risks
- available assets
- area of operations (AOO)
- criteria for success
- timeline
- force capabilities
- command and control structures

The analysis of the concepts involved in this process and the military planning models proposed in the literature facilitate the identification of relevant concepts to be incorporated into ontological models for courses of action or planning.

Moreover, the study of different scenarios enables the identification of more specific knowledge concepts that are to be examined during the planning process in a particular context.

3. Planning models

In this section, we present the most important planning models reported in the literature within the AI planning community, namely KRSL (Knowledge Representation Specification Language), CPR (Core Plan Representation), I-N-OVA (Issues-Nodes-Ordering/Variables/Auxiliary), SPAR (Shared Planning and Activity Representation), and JFACC (Joint Forces Air Component Commander).

3.1 History

The AI planning community has used explicit domain description languages and plan definitions for years (e.g., STRIPS representation of actions). In the 1990s, several initiatives began with the aim of building standardized models to facilitate the sharing of information about processes, plans and activities across organizations.

In 1992, under the DARPA/Air Force Research Laboratory Rome Planning Initiative (ARPI), a number of participants created Knowledge Representation Specification Language (KRSL). New insights led to an outline plan model called KRSL-Plans. Since 1995, there have been a number of initiatives to standardize terminology in the area of activities and plans: PIF (Process Interchange Format), NIST (National Institute of Standards and Technology) PSL (Process Specification Language) for manufacturing, and the Object Model Working Group's CPR (Core Plan Representation). Work on process models and grammars for describing the actions and products flowing from US JFACC (Joint Force Air Component Commander) (represented in Figure 1 as activity grammars) also contributed to ARPI.

Finally, SPAR (DARPA Shared Planning and Action Representation) [28] is an effort drawing from these previous works.

The evolution of these planning models is represented in Figure 1.

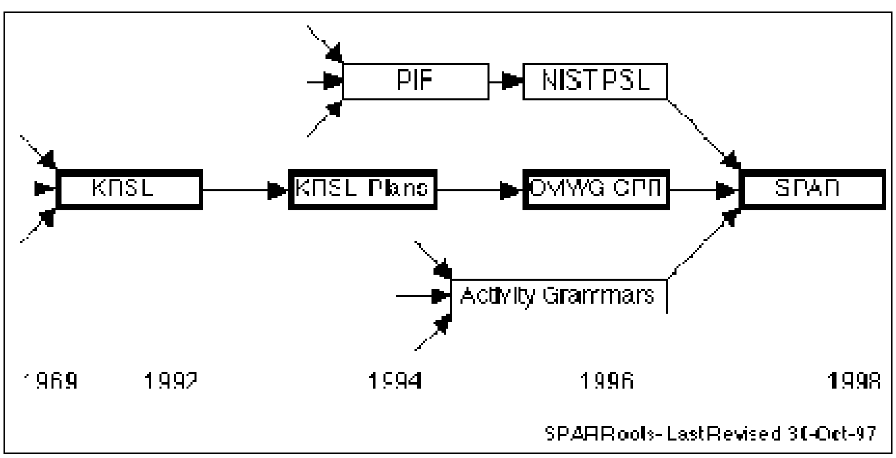

Figure 1. *History of planning models*

In the next several sections, we review some of the initiatives and models resulting from the ARPI program, namely KRSL/KRSL-Plans, CPR, SPAR, I-N-OVA, as well as JFACC.

3.2 KRSL/KRSL-Plans

KRSL (Knowledge Representation Specification Language) [23] was intended to provide a sharable ontology for representing plans and planning information as an interchange medium for ARPI systems and as a means of specifying shared domain information.

The ontology has two major aspects: an abstract ontology setting out major categories such as space, time, agents, actions, reasoning, and plans; and a set of modular specialized ontologies that augment the general categories with sets of concepts and alternative theories of more detailed notions commonly used in planning systems, such as theories of time points, temporal relations, and complex actions.

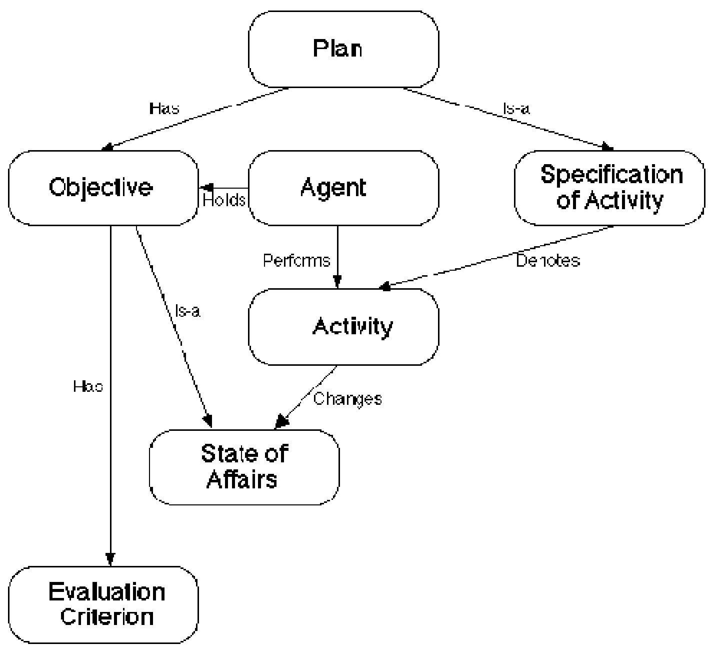

Figure 2. *KRSL-Plans high-level concepts*

KRSL-Plans defines the following entities (Figure 2):

- A *plan* is a *specification* of *activity* to meet one or more *objectives*.
- A *specification* of *activity* denotes or describes one or more *activities*.
- An *activity* may change the *state of affairs*.
- *A state of affairs* is something that can be evaluated as holding or not.
- An *agent* can perform *activities* and/or hold *objectives*.

- An *objective* may have one or more *evaluation-criteria*.
- An *evaluation-criteria* is an *aspect* of [past, present or possible future] *states-of-affairs* or an *aspect* of [one or more] *plans*. This is represented in Figure 2 through the *Objective* concept.
- An *evaluation* is a predicate (holds/does not hold) or a preference ranking based on *evaluation-criteria*.
- An *activity* takes place over a TIME-INTERVAL identified by its two ends, the BEGIN-TIME-POINT and the END-TIME-POINT. The BEGIN-TIME-POINT precedes the END-TIME-POINT.
- An *activity-specification* may have *activity-constraints* associated with it and a *time-interval*.
- An *activity-specification* may include one or more *sub-activities*.
- *Sub-activities* are the constituent activities designated in any *activity-specification*.
- *Primitive activity* is an *activity* whose *activity-specification* contains no (further) *activities*.
- *Activity constraints* can be stated with respect to none, one or more than one time-point. They express things that are required to hold and they can be evaluated with respect to a specific plan as holding or not holding.

Although this effort established a range of entities needed in a plan representation, the KRSL model was considered too rigid and lacking in much that was already being done in AI planning research, but it at least established a range of entities needed in a plan representation and influenced subsequent work.

3.3 CPR

Core Plan Representation (CPR) [12,13] is an effort to develop a plan ontology that supports the representation needs of many different planning systems in various military domains. It was developed in part for the Joint Task Force Advanced Technology Demonstration (JTF-ATD). The design of CPR is an attempt to unify the major concepts and advancements in plan and process representation into one comprehensive model.

3.3.1 Ontology content

The CPR model expresses information common to many plan, process and activity models. The core concepts that are common to all the efforts reviewed (e.g., KRSL, O-PLAN) are *Actor*, *Objective*, *Resource*, and *Action*. An *Action* is performed by an *Actor* in order to accomplish some *Objective*. The actor may utilize a *Resource*.

The CPR design can be created from the above concepts. A *Plan* consists of one or more *Actions* performed in pursuit of an *Objective*. *Time-spec* represents an association of the *Action* with time. Every *Action* has at least a beginning and an end, although it may be infinite or periodic. *Spatial Specifications* are also associated with *Actions* in absolute and relative terms.

Constraints are restrictions on some elements of the plan. For example, a proximity constraint among actions may specify that one Action must take place in a certain proximity to another Action, and a temporal constraint may specify that the end Time-spec of an Action must occur before the begin TimeSpec of another. A *Constraint* is an object that can be contained in any object of the *Plan*. Due to the aggregate nature of plans, it is appropriate to allow a *Plan* to be associated with another *Plan* (a plan being composed of subplans). The same argument is made for representing subobjectives and subactions.

During execution of the *Plan*, *Objectives* are reviewed to gauge the effectiveness of the *Plan* against a set of *Evaluation-Criteria*.

Further, *Domain objects* are used to represent entities referred to in an *Action*. *Effects* state how an *Action* is expected to change the state of affairs.

An *Event* represents something that is not part of the plan that has happened in the world. The *Inventory* keeps track of the current state of available resources.

World is the top-level object that aggregates other objects (plans, record of execution of plans, etc.). The world model contains a snapshot of the relevant features of the state of the world at a point in time or during an interval of time.

Plans must be able to represent incomplete information about the world. The classes *Uncertainty* and *Imprecision* represent two types of incomplete information that are both required. *Uncertainty* expresses the degree of confidence in information, whereas *Imprecision* represents the degree to which an exact value cannot be specified. Associated with *Uncertainty* is *InfluenceNetwork*, which keeps track of how uncertainties influence each other. The *Uncertainty* object holds a measure of the uncertainty associated with a piece of information. There are many ways to express an uncertainty measure: a Bayesian scheme might have a single real number, a Dempster-Shafer scheme two numbers, etc. The CPR design did not take into account this aspect.

3.3.2 Knowledge representation

CPR is a scheme for representing plan information that is independent of any particular method of plan construction. It was developed as an ontology and implemented as an object-oriented design. CPR can be viewed as an object-oriented presentation of SPAR. Figure 3 illustrates the CPR model.

The planning representation of KRSL presents some similarities to CPR; indeed, some individuals who participated to the KRSL effort contributed significantly to the CPR design. But compared to KRSL-Plans, which presents an ontology of informal sentences, CPR presents an object-oriented software design developed from an ontology.

CPR has been translated into different representation languages (e.g., Protégé, KIF-Ontolingua) and was integrated into the Teknowledge Cycorp Knowledge Base under DARPA HPKB.

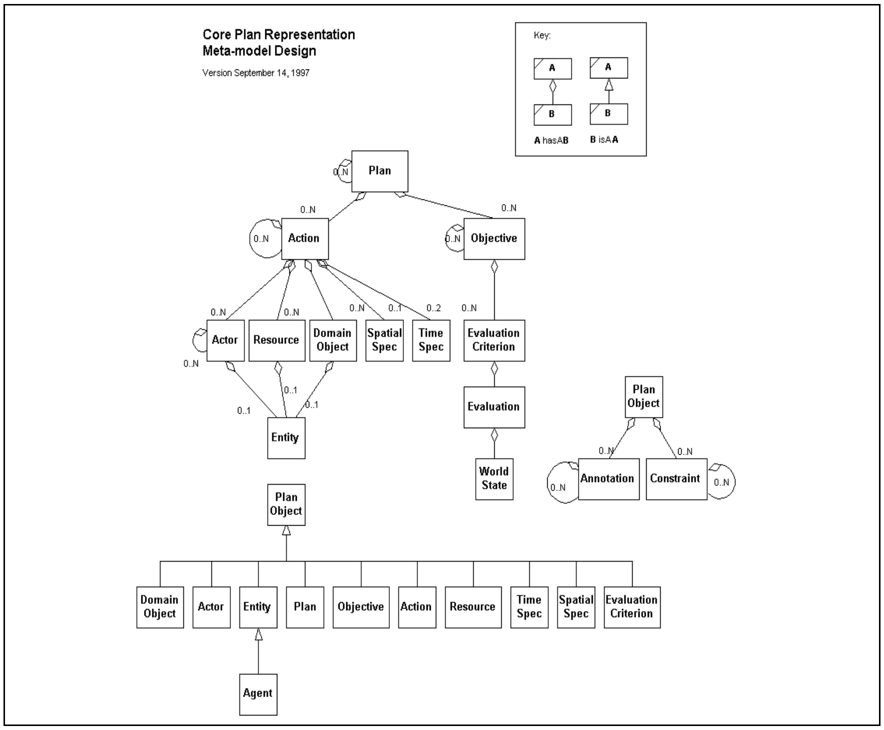

Figure 3. *CPR meta-model*

3.3.3 Ontology extension and use

The warplan [19] refers to an effort to specialize CPR for particular planning applications: intelligence planning, operations, and logistics.

An effort was undertaken to harmonize CPR with the *Air Campaign Planning* (ACP) grammar, and it presented several difficulties. First there was the challenge of harmonizing different paradigms of knowledge representation (Loom vs UML). Furthermore, the words used to denote concepts are often ambiguous. For example, the concept of Objective in ACP is the principal way in which commanders communicate with subordinates. Objectives specify what the subordinate is to do and what restrictions might be placed on how it is done (timing, location, resources). The CPR Objective concept is defined as “what is to be achieved” and is separate and distinct from the concept of Action, which specifies how to achieve the Objective. Actions have restrictions on time, location and resources. Examination of the concept in the two models revealed that the CPR Action was the same as the ACP Objective.

Harmonization also involved the extension of CPR with objects specific to ACP. The design was further specialized for an air force planning application (JFACC) by adding additional specific concepts to CPR.

For *Intelligence Planning*, significant additions include subclassing of Action to cover the five phases of intelligence operations: direction, collection, processing, production and dissemination. In addition, subclasses were created for resource types and roles. Roles are particularly important for the intelligence area. For that purpose, CPR follows a design choice of reifying roles when they are present (creating separate concepts that relate an entity to a role object).

For the *Logistics* application, significant specializations are an elaboration of Resource types: ConsumableResource, ReusableResource, SynchronouslyReusableResource, ExactCapacityResource, and NonSharableResource. Other specializations relate some real world entities to their roles. These include TransportationNode, Route, Transport and Driver-Pilot.

3.4 I-N-OVA

I-N-OVA [22, 24] is another effort that bears some similarity to CPR due to the significant contribution of the main researcher in the CPR effort. The <I-N-OVA> (Issues – Nodes – Orderings/Variables/Auxiliary) model provides a means to represent and manipulate plans or processes as a set of constraints. By providing a clear description of the different components within a plan, the model allows plans to be manipulated and employed in different environments from that in which they were originally generated.

The ontology is intended to act as a minimum core to which extensions can be added in a consistent, structured fashion. It is meant to support both the meta-level of reasoning about the activities involved in the planning process and the domain object level in which the plans are meant to be executed.

3.4.1 Ontology content

A plan is represented as a set of constraints that together limit the behaviour that is desired when the plan is executed. The set of constraints are of three principal types (Implied constraints, Plan Node constraints, and Detailed constraints), with a number of subtypes reflecting practical experience in a number of planning systems.

I: Issues (Implied Constraints)

N: Node Constraints (Activity Constraints)

OVA: Detailed Constraints:

- O: Ordering Constraints
- V: Variable Constraints
- A: Auxiliary Constraints such as:

Authority Constraints

Condition Constraints

Resource Constraints

Spatial Constraints

- Implied Constraints (or Issues): the pending or future constraints that will be added to the plan as a result of handling unsatisfied requirements, dealing with aspects of plan analysis and critiquing, etc. Implied constraints are the issues to be addressed, i.e., the agenda that can be used to decide how a plan is to be modified.
- Plan Entities or Plan Node Constraints: the main plan entities related to external communication of a plan. They describe a set of external names associated with time points. In an activity planner, the nodes are usually the actions in the plan associated with their begin and end time points. In a resource-centred scheduler, nodes may be the resource reservations made against the available resources with a begin and end time point for the reservation period.
- Detailed Constraints: Empirical work on the O-Plan planner has identified the desirability of distinguishing two special types of detailed constraints: Ordering or Temporal Constraints and Variable Constraints. Other Auxiliary Constraints relate to input (pre) or output (post) and protection conditions, resources, authority requirements, spatial constraints, etc.

3.4.2 Use of the ontology

The <I-N-OVA> model was used to characterize the plan representation used within O-Plan (Open Planning Architecture) [24]. The <I-N-OVA> work is related to emerging formal analyses of plans and planning. This synergy of practical and formal approaches can stretch the formal methods to cover realistic plan representations as needed for real problem solving, and can make improved analysis possible for production planning systems.

3.5 SPAR

In August 1997, DARPA and the Air Force Research Laboratory (Rome) Planning Initiative (ARPI) Program Managers proposed an effort to build on the accumulated expertise from past DARPA and ARPI sponsored research in order to create a shared plan representation suitable for use in ARPI and on applied research programmes in their communities. This effort was named Shared Planning and Activity Representation (SPAR) [20,21]. In addition to the past ARPI research efforts and experiences related to building shared plan representations (e.g., KRSL, KRSL-Plans, PIF, OMWG CPR), this work also intended to draw on relevant pre-standards research, including work on representing and interchanging process knowledge within and across enterprises.

The aim of SPAR is to produce the specification of a representation for plans, processes and activities for use by technology-base and applications projects in DARPA, Rome Laboratory and beyond, and to merge existing plan ontology work into a good, solid core representation as a starting point. The scope is to represent past, present and possible future activity and the command, planning and control processes that create and execute plans meant to guide or constrain future activity. These efforts are aimed at plan representations of a more general nature (Figure 4), and cover aspects of plan execution.

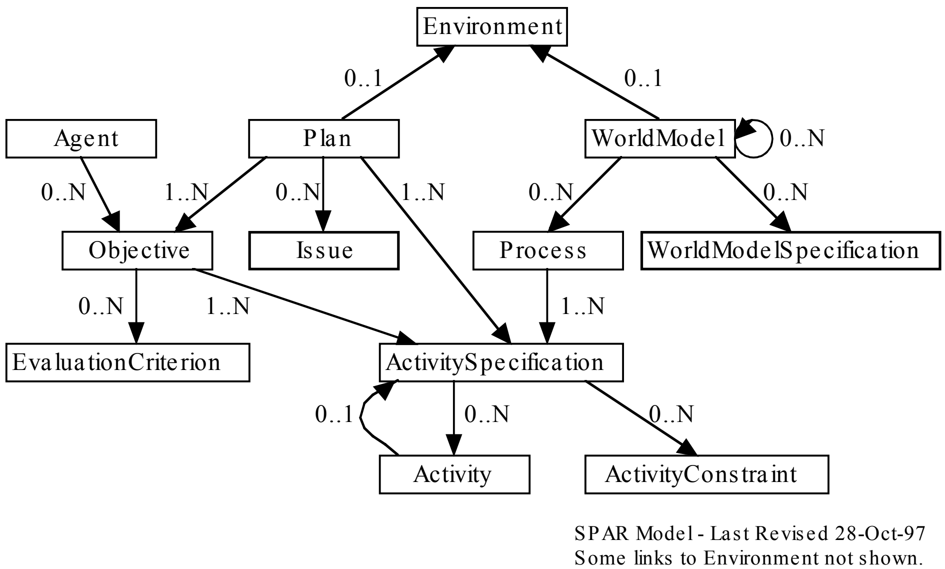

Figure 4. *SPAR model (initial version - 1997)*

A group of planning specialists (SPAR Core Group) contributed to the elaboration of the SPAR model. To that end, a set of representational and functional requirements had been assembled for the SPAR development process to help determine the scope and priorities of the project, elicit concepts and constructs, and gauge the adequacy of the SPAR representation.

The representational groupings define the elements that are needed to express information centred around plan representations. They included:

- activities
- agents
- control structures/execution/simulation
- domain knowledge
- evaluations
- general structures
- goals, requirements, objectives, mission, tasks

- organizational
- plans/schedule
- rationale
- resources/objects
- states
- time/space
- uncertainty/ambiguity

The functional requirements defined some of the intended uses of a rich, shared plan representation. These uses have been clustered into the following categories:

- communicating plans
- domain building
- organizational support
- plan editing/browsing
- plan execution
- plan/schedule generation
- plan evaluation/critique
- planning system synthesis
- task assignment

The KRSL-Plan description was used as a starting point for the SPAR model. The resulting model (see Figure 5) contains similar entities such as: a Plan relates an Activity-specification and Objective-specification. An Activity-Specification describes an Activity and can include Activity-Constraints. An Objective-Specification can include Objective Constraints that impose restrictions over a set of World States or which specify a required Activity. An Objective may have one or more Evaluation-Criteria, which may be applied to one or more World States to create an Evaluation. Execution of an Activity can change the World. An Agent is an Activity-Relatable-Object which can perform Activities and hold Objectives. An Activity takes place over a Time-interval identified by its two Timepoints (Begin-Timepoint, End-Timepoint).

The SPAR model provides some extensions [29].

- New terminology: action, actor, resource, subactivity, subobjective, process, and primitive-activity.
- World State: snapshot of the World. A World model provides a description of the World, and a World State description describes a set of World States (actual, expected or hypothetical).
- Timelines

- Values, Uncertainty, and Imprecision: an entity can have a set of properties; each Property has a set of possible values (Domain); a Property may be assigned one or more specific Values from its associated Domain; an asserted Property/Value assignment may have a Probability and an Imprecision.
- Conditions and Effects
- Evaluation details (Evaluation-Criteria, Evaluation-Result, Evaluation-Categories, Preference-Criteria).
- Libraries

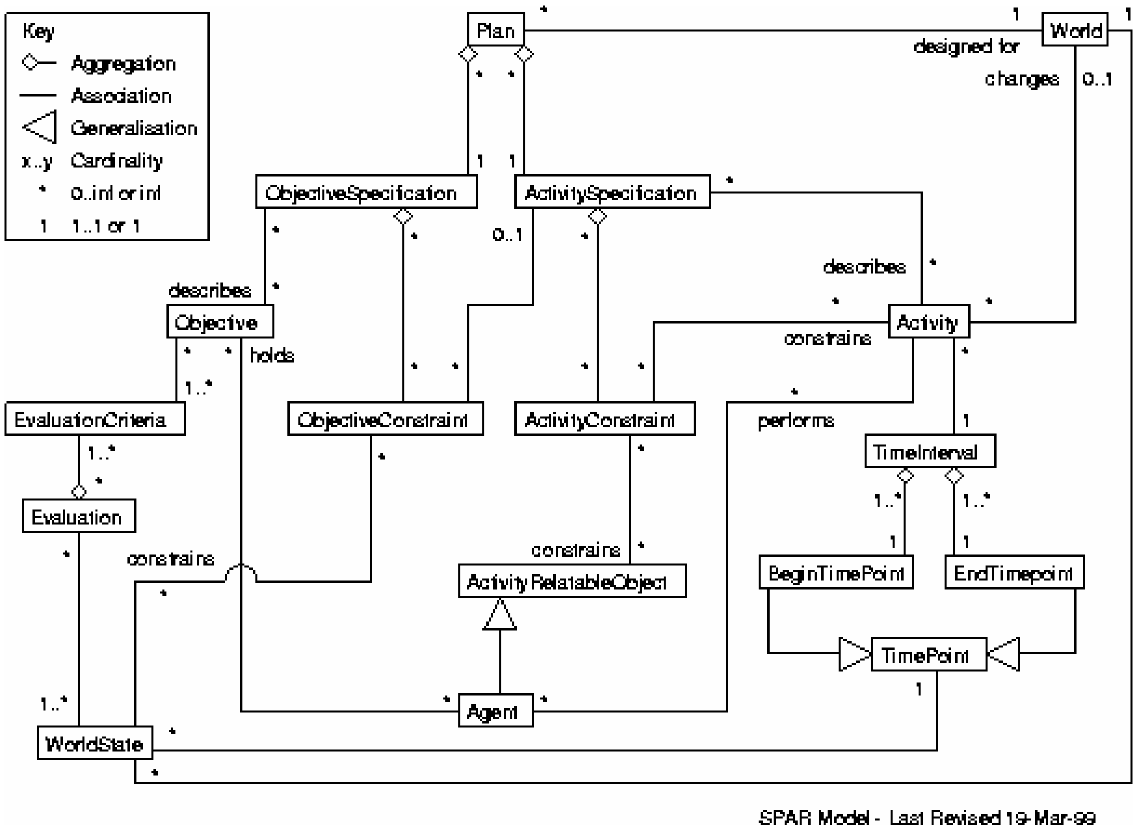

Figure 5. *SPAR Model (reviewed version - 1999)*

3.6 JFACC

The Joint Forces Air Component Commander (JFACC) ontology was developed by the Loom group from the Information Science Institute of the University of Southern California (ISI USC), as part of the JFACC program sponsored by DARPA and AFRL [9]. In parallel, the Expect group (from ISI USC), led by Yolanda Gil, developed a plan evaluation tool called INSPECT [26].

3.6.1 Context

The JFACC ontology was developed for the domain of air campaign planning. A primary objective of the JFACC program was to semi-automate the process of constructing computer models that represent military objectives, planned courses of action, etc. Ontologies provide conceptual building blocks for these models and knowledge bases capture the semantics of real-world entities referenced in a model. Furthermore, there were several goals:

1. Facilitate inter-operation and communication between systems by providing a common terminology.
2. Promote sharing of knowledge between systems.
3. Create a repository of general knowledge about air campaign planning that could be used across different applications.

Ontologies developed for the ARPA Rome Planning Initiative (ARPI) were analyzed and integrated into a common ontology for ARPI and JFACC. The JFACC ontology draws from the ACP-Sensus ontology, the INSPECT Air Campaign Objectives ontology and domain model, as well as ARPI Planning and Scheduling ontologies and PIF process ontology. The JFACC knowledge base contains knowledge about air campaign planning processes, planning factors, available assets and their capabilities, generic tasks, strategies and objectives. In this context, some problems related to ontology construction were addressed:

- How to translate and import a publicly available ontology written in another knowledge representation formalism;
- How to merge the contents of two existing, independently developed knowledge bases into an ontology;
- How to structure a large ontology into reasonably independent modules;
- How to import knowledge from an ontology into several applications, and how to extract part of the knowledge from the ontology to use in a knowledge-based application that needs the knowledge in another format.

3.6.2 Ontology content

The Loom team tried to use as much as possible the formally described knowledge that was already available about the domain: existing knowledge bases, theories of fundamental concepts such as *time* and *system*, general ontologies of domain elements (e.g., aircraft, weapons), as well as existing ontologies available from public repositories.

The development of the JFACC ontology has been pragmatic as a function of the needs of the applications that make use of it. Some concepts are rigorously specified (e.g., time, objectives, plans), whereas other concepts (e.g., action, space) are represented in a simplified manner.

3.6.2.1 Ontology import

The JFACC ontology reused the knowledge base of the INSPECT system, which included detailed representations for all the main elements of air campaign plans: campaigns, objectives, missions, phases, areas, sequencing, etc. It also included an extensive topology of military targets, from military headquarters to petroleum production facilities, and basic representations for resources such as aircraft and weapons.

Two ontologies of aircraft that explored different views on the domain (one built from Fact Sheets from the US Air Force, and the other from the INSPECT project) were merged to benefit from both existing ontologies and to create a more comprehensive domain model. This was a tricky process because of the structural differences between the two ontologies, but the positive aspect was the integration of both views. The resulting ontology is presented in Figure 6.

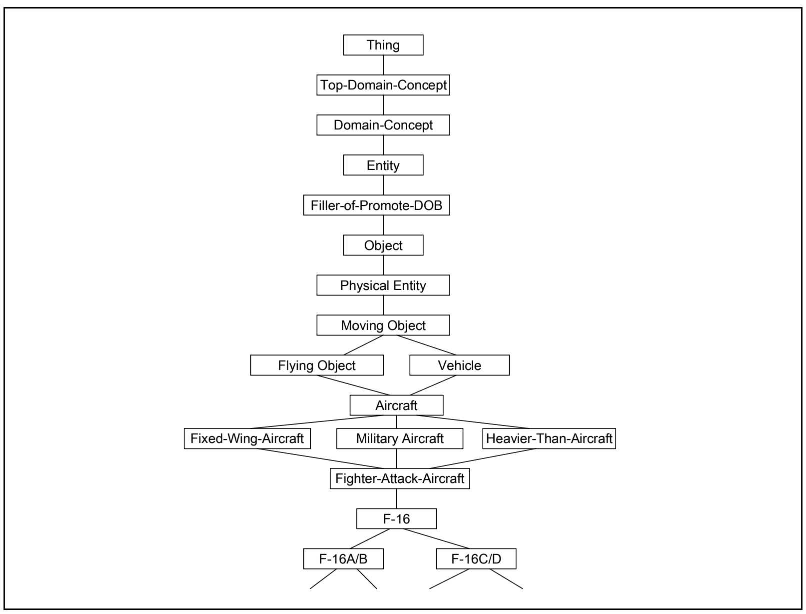

Figure 6. *The JFACC ontology (top-level view)*

The JFACC ontology required a representation of some of the concepts in a more principled way. In particular, a complete and well-founded ontology of time and systems was incorporated. Time is an essential constituent of the air campaign domain. Thus, an existing ontology of time based on Allen's theory of time was reused (with some translation mechanisms required). Many temporal relations were included in this ontology.

3.6.2.2 Ontology structuring

The JFACC ontology was modularized to make its maintenance easier and its construction more rational. It was a necessary process to manage the contents of an ontology with more than a few hundred entities. An aspect of the modularization process is to identify the dependencies between entities from different sections of the ontology. The JFACC researchers used the Loom contexts to obtain modularization. A Loom context contains a symbol table that maps a collection of names onto corresponding associated logical entities (relations and individuals). Each context inherits zero or more other contexts. If a context C2 inherits a context C1 then all names and entities belonging to C1 also belong to C2. Figure 7 illustrates the inheritance lattice for the JFACC ontology. Each node is considered to define its own subontology, so ontologies are mapped one-to-one onto contexts. The ability to reference names defined in a context not inherited by a given context (e.g., cross-references between *aircraft*, *weapons* and *fuel* contexts) was realized through the construction of an artificial context (the *entities* context) that holds general versions of concepts visible to each of these other contexts. For example, the general concept *aircraft* is defined in the *entities* context so that it can be referenced by the *weapons* and *fuel* contexts, but the definitions for types of aircraft (e.g., fighter aircraft) are in the *aircraft* ontology.

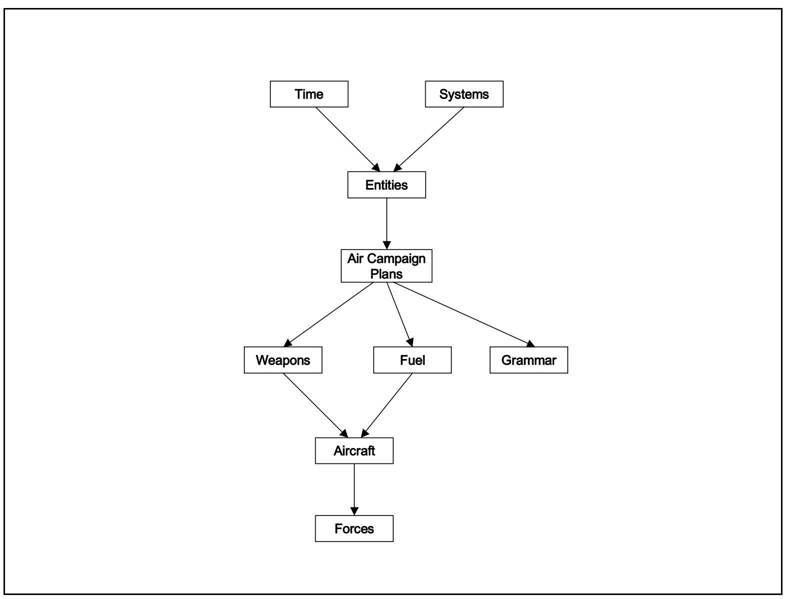

Figure 7. *Organization of the JFACC ontology into its constituent modules*

In the figure, an arrow indicates that one ontology is based on another. For example, the *Entities* ontology imports both *Time* and *Systems*, and is imported by *Air Campaign Plans*. The individual module contents are described in the table below.

Table 1. Ontology content

Name of the ontology	Content
Systems	Defines general systems, their decomposition into subsystems and primitive components, their inputs and outputs, etc. It was useful to model enemy systems varying from distribution to transportation networks to military systems for command and control
Time	Definitions for time points and intervals, dates (absolute and relative), and many relations between time points and intervals.
Entities	Micro-upper-level ontology that encapsulates the definition of general types of entities that are shared among many other component ontologies. It contains also extensive hierarchies of types of targets, military actions, action capabilities, and abstract characteristics of the situation or state that are used frequently when defining air campaign plan objectives.
Air Campaign Plan	Overview of the basic elements that characterize air campaign plans, such as campaign, scenario, participants, commanders, plans, phases, objectives, etc.
Weapons	Definitions and extensive information about missiles, guns, bombs, and munitions.
Fuel	Definitions and data for the main types of fuel and additivies used by US military aircraft.
Aircraft	Knowledge about types of aircraft in the US military, including data about engines, pods and fuel tanks these aircraft can carry.
Forces	Definitions about military force units, including definitions for military bases and installations. Facilitates reasoning about logistics.
Grammar	Used to define objectives in a structured manner, using a syntax-oriented editor. The ontology contains the concepts and relations necessary to map the representation of objectives in the air campaign plans ontology into the grammar representation.

The final version of the ontology contains 1750 entities (about 1100 concepts, 400 relations and 250 instances).

3.6.3 Knowledge representation

The JFACC ontology is represented in Loom, a knowledge representation framework based on Description Logics that provides a semantic network approach to knowledge representation. Concepts are defined with roles or slots used to specify attributes of the concepts. The semantics of the representation language are very precisely specified, making it possible to build a classifier.

The JFACC ontology was built and managed by the Ontosaurus ontology server based on the Loom knowledge representation system.

3.6.4 Use of the ontology

The ontology has been used in several applications, namely the INSPECT plan critiquer, the Mastermind Plan Editor and the Strategy Development Assistant.

The Strategy Development Assistant (SDA) is a knowledge-based, mixed-initiative planning system designed to support air campaign planning. It supports military planners to decompose high-level objectives into more specific subobjectives. The SDA assists the user in this decomposition process by providing suggested decompositions based on the current situation and high-level goals.

The SDA was created after the JFACC ontology was already in place and its construction demanded less knowledge acquisition and modelling than previous applications. However, using an ontology in several applications is not a trivial task, and requires a translation process equally difficult to the one used in importing an ontology. Moreover, it was found difficult to structure an ontology used in different applications. A solution is to maintain a core representation and layer additional information necessary to make specific uses of the knowledge in this core.

Finally, the application showed that the use of an ontology substituting the usual database or object-oriented schema offers a semantically richer model of the domain.

4. The DISCIPLE project

This section describes the DISCIPLE project conducted at George Mason University and its applications to military planning domains: courses of action evaluation (DISCIPLE-COA) and centre of gravity determination (DISCIPLE-COG).

4.1 Introduction

The DISCIPLE project is part of the HPKB (High Performance Knowledge Base) DARPA program and has been conducted by the Learning Agents Laboratory (LALAB) at George Mason University. The project was pursued as part of the RKF (Rapid Knowledge Formation) program (Disciple-RKF).

The project started with the design of the Learning Agent Shell (Disciple-LAS) [6] that was used in several applications: Disciple-COA, Disciple-Workaround and Disciple-COG. The project Disciple-LAS is an integrated set of modules for rapid development of practical end-to-end knowledge-based agents, by domain experts, with limited assistance from knowledge engineers [5]. Central to the approach is an architectural separation of the agent's knowledge base into two main components, an ontology that defines the concepts from the application and a set of problem-solving rules expressed in terms of these concepts.

The process of building the knowledge base of a Disciple agent consists of three main steps:

1. A *modelling step*, where the expert and the knowledge engineer define at a conceptual level how the agent will perform its tasks. This step results in an informal specification of the concepts needed to be represented in the agent's ontology.
2. An *ontology creation step*, where the knowledge engineer and the expert import some of the concepts identified in the previous step from existing repositories of knowledge, and define the rest of the concepts. This step results in an initial knowledge base that contains an incomplete ontology and no rules.
3. A *teaching step* where the subject matter expert teaches the agent how to perform its tasks in a way that resembles how the expert would teach a human apprentice when solving problems in cooperation. During this step the agent will learn problem-solving rules from the expert, and will also extend and update the ontology.

This process identifies the concepts that are needed in the ontology in order for the agent to perform its tasks. This is in contrast to other approaches to ontology construction, such as that used with the large knowledge base Cyc [16], where the goal is to develop a general ontology that tries to abstract away from the specifics of a particular application domain. In the Disciple approach, the goal is to create a domain-specific ontology that is very well suited to solving problems in the application domain concerned. An important aspect of Disciple is that the ontology itself is evolving during knowledge acquisition and learning.

With Disciple, the ontology serves as the generalization hierarchy that is used to learn general rules from specific examples. Furthermore, the representation of the ontology is based on the OKBC (Open Knowledge Base Connectivity) knowledge model to facilitate knowledge import from OKBC-compliant knowledge servers [10].

The architecture of the Disciple environment is presented in Figure 8. It includes an import/export module to acquire external ontology components (KIF, OKBC), as well as several knowledge object editors (object, task, feature, rule). These components are integrated by the Disciple knowledge base manager so that the problem solver can exploit the different knowledge elements.

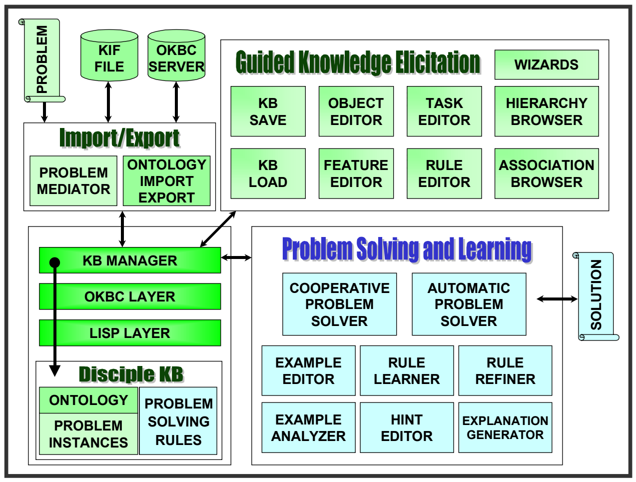

Figure 8. *Architecture of Disciple*

4.2 DISCIPLE-COA

One of the DARPA HPKB challenge problems was to construct a critiquing agent that could evaluate military courses of action for ground combat operations, i.e., identify its strengths and weaknesses with respect to the principles of war and tenets of Army operations. To address the HPKB course of action challenge problem, the Disciple architecture was extended and used to develop the Disciple-COA agent.

4.2.1 Knowledge representation

The ontology consists of hierarchical descriptions of objects, features and tasks, represented as frames, according to the knowledge model of the Open Knowledge Base Connectivity (OKBC) protocol [10].

The *objects* represent either specific individuals or sets of individuals in the application domain. They are hierarchically organized according to the generalization relation (subclass-of/superclass-of and instance-of/type-of). In this context, an object may have more than one parent. The hierarchy of objects is used by the Disciple agent as a generalization hierarchy, which is one way to generalize an expression by replacing an object with a more general one from such a hierarchy.

The *features* are used to further describe objects, other features and tasks. Characteristics of features include their domain (the set of objects that could have this feature), their range (the set of possible values of the feature), and functions for computing their values. Features are also hierarchically organized. The feature generalization hierarchies are extensively used in analogical reasoning.

Finally, a *task* is a representation of anything that the agent may be asked to accomplish. Tasks are organized according to the generalization relation. The hierarchies of tasks are also used in analogical reasoning.

4.2.2 Ontology content

The Disciple Knowledge Base is composed of six types of knowledge elements: objects, features, tasks, examples, explanations and rules. Among these, the objects, features and tasks form the agent's ontology.

The input ontology contains the terms needed to represent the COAs to be critiqued, and was shared by all the developed critiquers. The top level of the object ontology (upper level) includes the concepts for representing *geographical information, military organizations* and *equipment, descriptions of specific COAs, military tasks, operations* and *purposes*. Each of these concepts is the top of a specialized hierarchy, such as the hierarchy of organizations shown in Figure 9. The leaves of this hierarchy are specific military units, corresponding to a specific COA to be critiqued by Disciple.

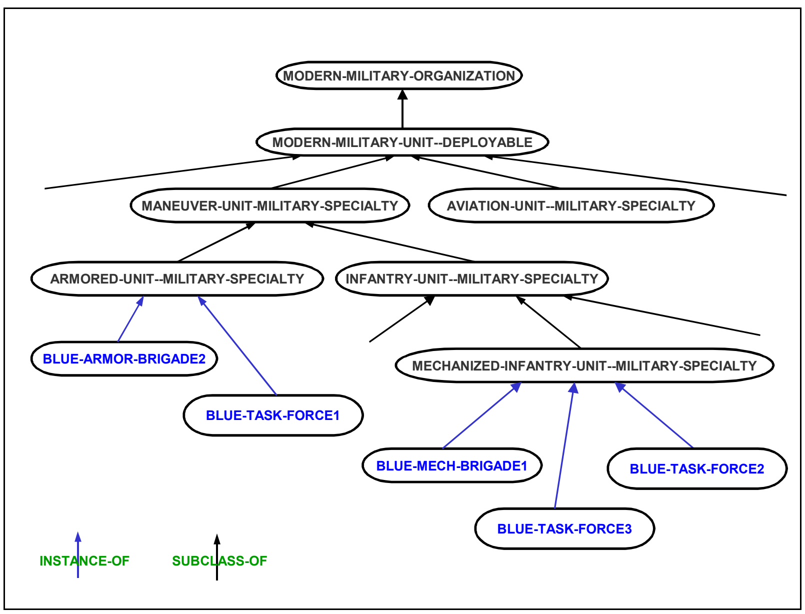

Figure 9. *Fragment of the military organization ontology*

The objects represent either specific individuals or sets of individuals in the application domain. The objects are hierarchically organized according to the generalization relation (subclass-of/superclass-of and instance-of/type-of). Figure 10, for instance, presents a fragment of the military task ontology used to model the COA analysis domain and a particular instance.

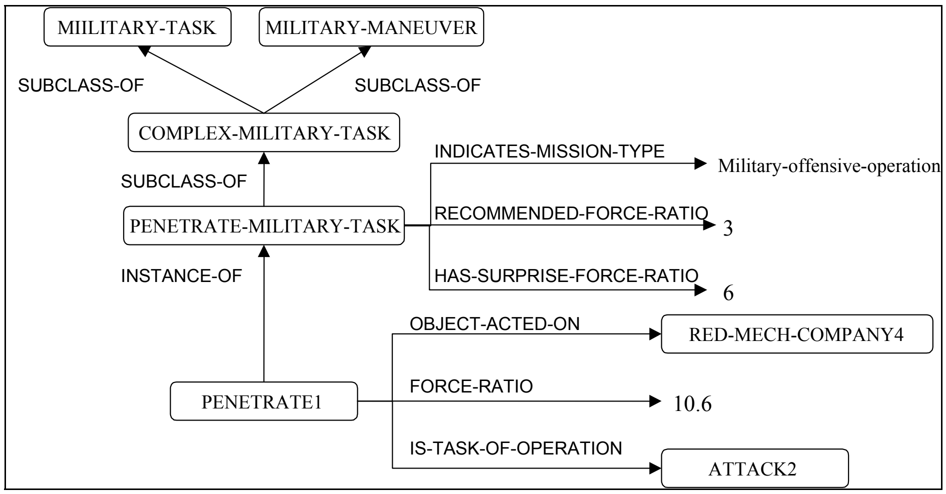

Figure 10. *Fragment of the military task ontology*

Each concept and instance of the object hierarchy is described by specific features and values. For instance, the bottom part of Figure 10 shows the description of the specific military task PENETRATE1 that is defined as being a penetration task, and therefore inherits all the features of the penetration tasks, in addition to the features that are directly associated with it.

The Disciple-COA ontology contains 801 concepts, 444 objects and task features, 360 tasks and 342 rules. Also, the description of a course of action was represented with around 1500 facts.

4.2.3 Ontology import/export

For Disciple-COA, an initial ontology was defined by importing the ontology built by Teknowledge/Cycorp [16] for the courses of action challenge problem, which contained the vocabulary needed to represent courses of action. The Disciple-COA ontology was first translated from CYC's language into KIF (Knowledge Interchange Format), and from there it was translated into the representation language of Disciple and the other critiquers. The imported ontology was refined and extended for the Disciple-COA agent using the ontology building tools of Disciple. One example is the military task substructure. It is noteworthy that the Disciple knowledge base for COA critiquing has been translated into a CYC microtheory.

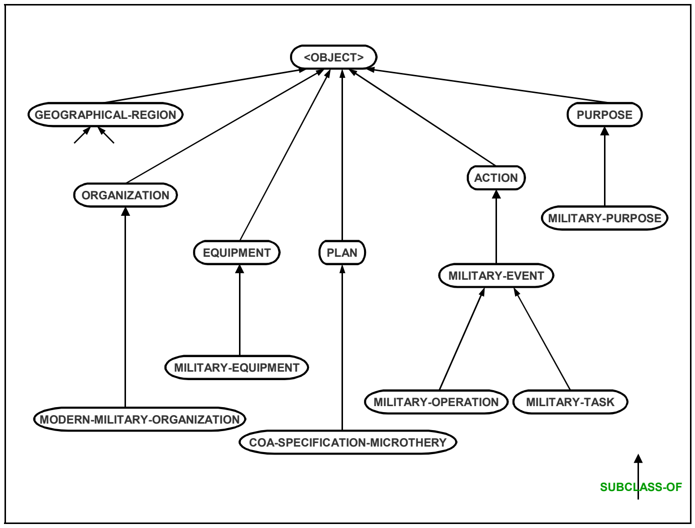

Figure 11. *Ontology fragment imported from CYC for Disciple-COA*

4.2.4 Development methodology

To build a knowledge base, the first step is to develop a model of the application domain that will make explicit at a qualitative and informal level the way the expert solves problems. In the case of Disciple, this means modelling the process of solving a specific problem as a sequence of qualitative and informal problem reduction steps, where a complex problem is successively reduced to simpler problems. Disciple uses its task reduction rules to reduce the current task to simpler tasks, showing the expert the reduction found. Then, experts build an ontology that will define the terms used to express the problems and their solutions. After that, they define formal problem-solving rules to verify the rules and update them [3].

As mentioned above, the process of building the knowledge base consists of three main steps: domain modelling, ontology creation and rule learning. For example, an assessment task is successively reduced to simpler assessment tasks and ultimately reduced to assertions on how the course of action conforms to the principle of objective. From an ontology development point of view, this process is important because it informally identifies the concepts that are needed to be present in the ontology. To "Assess the course of action according to the principle of objective", one has to consider the features that characterize the objective, and these are specified by the subject matter expert as being "identification", "attainability", and "decisiveness". Therefore, the current task is reduced to the following simpler assessment tasks: "Assess identification of objective", "Assess attainability of objective", "Assess decisiveness of objective". Assessing the attainability of the objective is applicable for the main effort and an offensive mission. Therefore, the ontology has to contain a classification of COA missions into offensive missions and defensive missions, and each specific course of action would need to have a feature that will specify its mission type. Through this process, the elements that need to be represented in the ontology result from an analysis of the agent's task reduction process in a particular application domain.

An experiment took four military professionals experienced in both the tactical and operational levels of war but having no prior knowledge engineering experience, and gave them approximately sixteen hours of training in AI and the use of Disciple-COA. Then starting with a knowledge base containing the complete ontology of objects and features in Disciple-COA but no rules, the military professionals were asked to train the agent to critique courses of action based on two principles of war – offensive action and security.

4.2.5 Ontology development tool

The imported ontology was developed using the ontology building tools of Disciple: object, feature, task and rule editors and browsers (see Figure 12). Furthermore, ontology management assistants guide and support the user in modifying the knowledge base so that the ontology always remains in a consistent state.

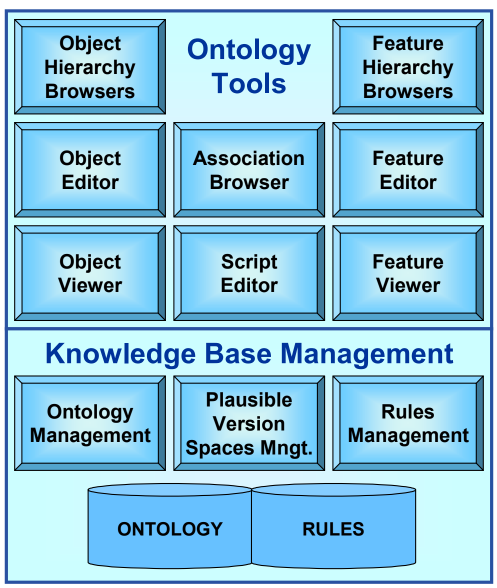

Figure 12. *Ontology management tools in Disciple*

4.3 DISCIPLE-COG

Disciple-COG is part of the DARPA Rapid Knowledge Formation (RKF) program. In 2000-2001, an intensive effort was made by researchers from GMU LALAB and US Army War College to extend the Disciple-COA ontology to represent the military concept of centre of gravity used at the strategic and operational levels of war [7, 29].

This was intended to help the design of intelligent agents that assist in strategic centre of gravity determination. The WWII Okinawa and Sicily campaign scenarios served as the initial examples.

4.3.1 Centre of gravity

The most often quoted definition of the centre of gravity (COG) concept was originally introduced by Carl Von Clausewitz in [On War, 1832]. "The centre of gravity of an entity (state, alliance, coalition or group) is the foundation of capability, the hub of all power and movement, upon which everything depends, the point against which all the energies should be directed".

The concept of COG is of primary importance in the military planning process. At the strategic level of war, COG determination for friendly and opposing forces is essential for maintaining focus on strategic goals, for allocating and using military resources, and for winning the war [7].

4.3.2 Ontology development and content

A group of subject matter experts from the US Army War College was formed and produced a monograph to give a definition of the concept based on the Clausewitz theory. Elements to be taken into account in the COG determination process include overall organization, scenario, forces and goals, economic factors, geographic factors, military factors, political factors, governing and controlling elements, civilization factors, and others. The high-level concepts for the COG ontology are presented in Figure 13.

The ontology development for the COG application built upon what was done previously for the course of action application. The expansion of this ontology for COG determination and analysis is extensive and complex because of the much greater coverage required for the new domain. In just one category, "Economic Factors" for example, there are a plethora of objects, features, tasks and rules that can be added to the ontology used in Disciple-COA. Thus, new concepts such as GROSS-DOMESTIC-PRODUCT, ECONOMIC-DEPENDENCY-ON-IMPORTED-FOSSIL-FUEL, ELECTRICAL-PRODUCTION-CAPABILITIES, and more are also needed.

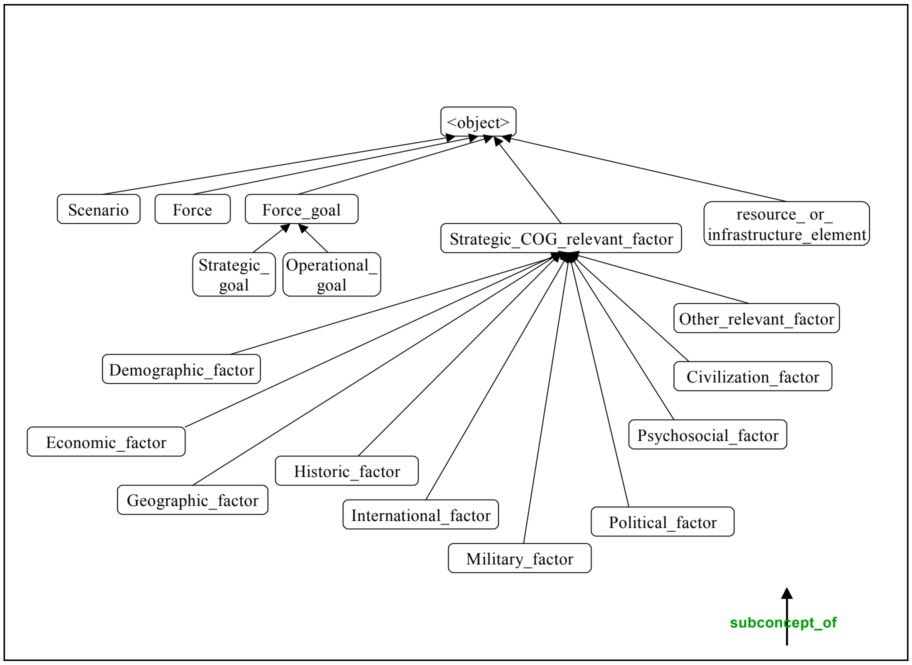

Figure 13. *High-level concepts of COG ontology*

A key first step in successfully conducting COG determination and analysis as described in the COG Monograph is the correct selection of the opposing force to be analyzed as well as a detailed identification of the opposing force's composition, characteristics and nature. Is the opposing force a coalition or alliance of nation states or an individual nation state? Is it an ad hoc group of clans or a single terrorist organization? Figure 14 is a sample of the high-level concepts and relationships necessary for correctly representing opposing forces and some historical instances of such forces.

Each of the high-level concepts of the COG ontology was refined to incorporate relevant concepts to be taken into account in the COG determination process.

Among the elements to be represented in the COG ontology, the most difficult challenge encountered in this effort was the closely related areas of "Primary Controlling Elements" and "Type of government". From the monograph, a recommendation was made to extend the ontology to cover religion-focused nation states and groups.

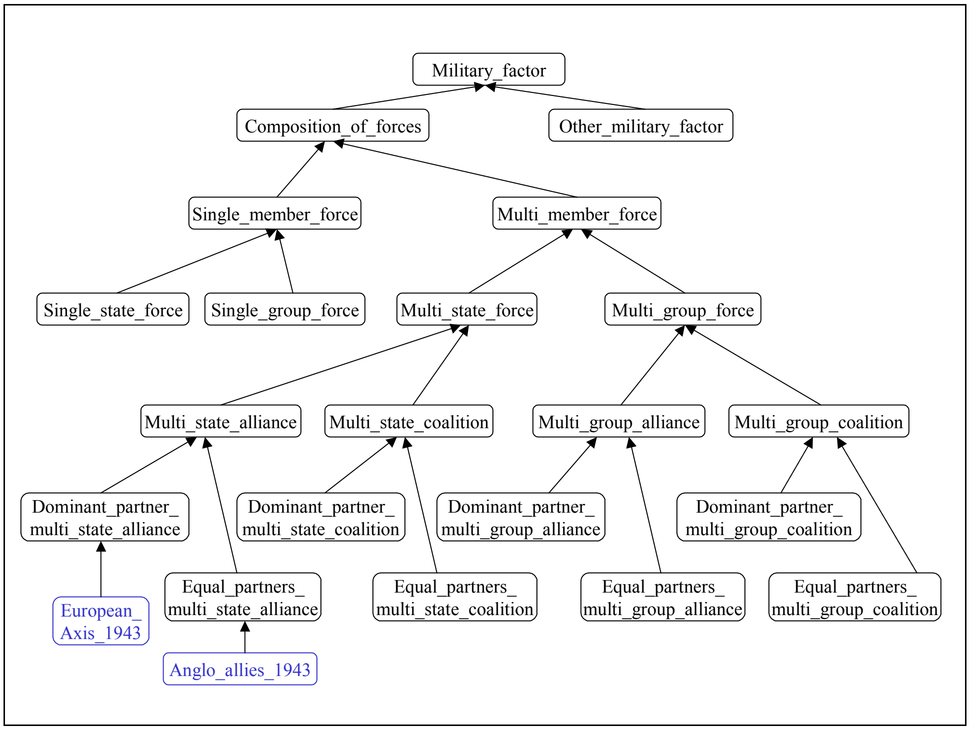

Figure 14. *Military Forces for COG*

The developed ontology is documented in a set of diagrams (over 100 pages in length) that represents the concepts and relationships expressed in the COG Monograph with instances of those concepts and relationships evident in the Okinawa and Sicily campaigns.

4.3.3 Ontology Import / export

When a knowledge engineer works with a subject matter expert to define an initial domain model, they identify the types of objects and features that are needed in the knowledge base. These objects and features focus the process of importing relevant ontological knowledge from existing knowledge repositories.

The ontology import module of Disciple is composed of three phases:

1. mixed-initiative retrieval of potentially relevant ontological knowledge from an external knowledge repository;
2. automatic translation of the retrieved ontological knowledge into an intermediate Disciple ontology;
3. mixed-initiative import from the intermediate Disciple ontology into the final Disciple ontology.

In general, one of the practical difficulties encountered in ontology import is the fact that the subject matter expert has to deal with the additional representation system and tools of the knowledge repository from which knowledge has to be imported. To alleviate this problem, for each knowledge repository from which they are importing knowledge in Disciple, a standard ontology retrieval interface is developed. This interface allows the subject matter expert to retrieve relevant knowledge from different representation systems without dealing with the tools or representation of that knowledge repository.

In the Disciple architecture there are three implementations of the standard interface, one for the CYC system, which already exists, one for any OKBC-compliant knowledge repository, such as Loom, Ontolingua, or Protégé, and another for older Disciple repositories.

The representation of the Disciple object ontology based on the OKBC knowledge model is usually less powerful than the representations of the knowledge servers from which they need to import knowledge. But the purpose is to import the relevant knowledge that can be represented in the Disciple object ontology in order to serve as a generalization hierarchy for learning of problem-solving rules.

In the second phase of the ontology import process, the retrieved CYC ontology is automatically translated into an intermediate Disciple ontology by a general rule-based translation engine that uses a CYC-Disciple rule translation library. Additional rule translation libraries need to be defined for each type of knowledge repository (e.g., for an OKBC-compliant knowledge server, for older Disciple repositories, etc.).

After the Disciple knowledge base has been developed by the expert using the teaching mechanism, the knowledge base is exported back into CYC as a separate CYC microtheory. There is no translation problem since CYC's knowledge representation is more powerful than that of Disciple. The new microtheory can be semantically incorporated into the CYC knowledge repository.

4.3.4 Ontology development and use

The resulting expanded ontology is exploited by DISCIPLE agents that assist in the identification of strategic COG candidates for each opposing force and members from the description of a scenario. The object ontology provides a representation vocabulary that is used in the description of the reduction and composition rules.

The scenario elicitation module allows an SME to create and update a scenario. The idea is to associate elicitation scripts with the concepts from the input ontology. The script associated with a concept plays multiple roles: it specifies how an instance of that concept is created, what features of the instance need to be elicited, how the dialog with the user takes place, and what graphical components are used in the dialog. To elicit a specific feature, Disciple also uses the information from the ontology about that feature, such as its possible values and cardinality.

5. PLANET

The PLANET[1] ontology was developed at the Information Sciences Institute of the University of Southern California [8] under the DARPA High Performance Knowledge Base (HPKB) program [2]. It leverages from previous work on the design of several plan evaluation systems. It is a middle-level theory that represents planning knowledge.

PLANET describes a reusable knowledge base of general principles about plan evaluation and critiquing. It includes ontologies to represent plans and critiques, an ontology of evaluation criteria, and problem-solving knowledge about how to evaluate plans according to those criteria. PLANET contains general, domain-independent definitions that are common and useful across planning domains. Its plan representation can be reused to build new applications by adding domain-dependent extensions to the general definitions and classes.

PLANET was used as a knowledge modelling tool to design representations of courses of action with a view to developing a system for COA analysis for the DARPA HPKB Battlespace Challenge Problem.

5.1 Ontology content

The PLANET ontology explicitly represents planning problem contexts as well as goals, actions, tasks and choice points. Each of these concepts is described below and illustrated in Figure 15.

A *planning problem context* represents the initial, given assumptions about the planning problem. It describes the background scenario in which plans are designed and must operate on. This context includes the initial state, desired goals and external constraints.

A *world state* is a model of the environment for which the plan is intended. A certain world state description can be chosen as the *initial state* of a given planning problem, and all plans that are solutions of this planning problem must assume that initial state.

The *desired goals* express what is to be accomplished as a result of executing the plan.

A distinction is made between *external constraints* imposed on planning problems and the *commitments* made by the plan.

External constraints may be specified as part of the planning context to express desirable or undesirable properties or effects of potential solutions to the problem, including user advice and preferences (e.g., the plan accomplishes a mission in a period of seven days, the plan does not use a certain type of resource, or transportation is preferably by tracked vehicles). The initial requirements expressed in the planning problem context need not all be consistent and achievable (since actions and tasks are subtypes of goal specification, as well as posted goals and objectives).

A *goal specification* is used to represent both capabilities and effects of actions and tasks as well as posted goals and objectives. *State-based goal specifications* are goal specifications that typically represent goals that refer to some predicate used to describe

[1] PLAN semantic NET

the state of the world, for example 'achieve (at JimLAX)', 'deny (atRed-BrigadeSouth-Pass)' or 'maintain (temperature Room5 30)'. *Objective-based goal specifications* are goal specifications that are typically stated as verb- or action-based expressions, such as 'transport brigade5 to Riyadh'.

Planning levels can be associated with task descriptions as well as goal specifications. Some AI planners assign levels to tasks, others assign levels to particular predicates or goals (e.g., ABSTRIPS). Levels are also used in real-world domains, for example, military plans are often described in different levels according to the command structure, echelons, or nature of the tasks.

Plan-task descriptions are the actions that can be taken in the world state. They include templates and their instantiations, and can be abstract or specific. A plan-task description models one or more *actions* in the external world. A *plan-task* is a subclass of *plan-task description* and represents an instantiation of a task as it appears in a plan. It can be a partial or full instantiation. A *plan-task template* denotes an action or set of actions that can be performed in the world state. Plan-task descriptions have a set of preconditions, a set of effects and a capability, and can be decomposed into a set of subtasks. The *capability* describes the goal for which the task can be used. A *precondition* represents a necessary condition for the task, and *effects* result from the execution of a task. Tasks can be decomposed into *subtasks*, which are themselves task descriptions.

A *plan* represents a set of commitments to actions to be taken by an agent in order to achieve specified goals. Several subclasses of plans are proposed: *feasible plan* (candidate plan), *justified plan, consistent plan*, and *complete plan*. It can be useful to state that a plan can form a *subplan* of another one. For example, military plans often include subplans that represent the movement of assets to the area of operations (i.e., logistics tasks), and subplans that represent a portion of the overall operation (i.e., force application tasks).

Possible *choices* among alternatives during the planning process are represented as commitments. *Plan commitments* are commitments on the plan as a whole, and may be in the form of actions at variously detailed levels of specification, orderings among actions and other requirements on a plan such as a cost profile. The tasks that will form part of the plan are represented as a subset of the commitments made by the plan. An ordering commitment is a relation between tasks such as (Before A B). A temporal commitment is a commitment on a task with respect to time, such as (Before ?task ?time-stamp).

The PLANET ontology also represents the context, assumptions and situation in which the plan is supposed to work in this domain. A COA is supposed to accomplish the mission and other guidance provided by the commander, and to work in the context of the given situation as analyzed by the commander's staff, which includes terrain information and enemy characteristics. A COA problem context is defined as a subclass of planning-problem-context, and its scenario is composed of commander products and staff products. All COA problems are attached to this problem context.

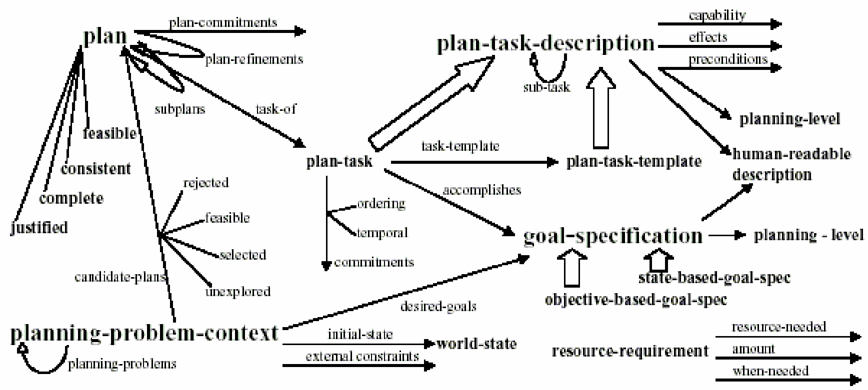

Figure 15. *Planet ontology*

5.2 COA example

An example of part of a COA textual statement follows:

On H-hour D-day, a mechanized division attacks to seize OBJ SLAM to protect the northern flank of the corps main effort. A mechanized brigade attacks in the north, as an economy of force, to fix enemy forces in zone denying them the ability to interfere with the main effort's attack in the south. A tank-heavy brigade, the main effort, passes through the southern mechanized brigade and attacks to seize the terrain vicinity of OBJ SLAM denying the enemy access to the terrain southwest of RIVER TOWN. [...]

A typical COA includes the overall mission and a set of tasks that need to be performed divided into five categories: close, reserve, security, deep and rear. The five task categories are represented as sub-plans (they are not subtasks or subgoals but useful categories for grouping the unit's activities). The close statements always contain a main effort for the COA and a set of supporting efforts. In PLANET representation, the mission defines two important features of the plan: its top-level goal (e.g., *protect the northern flank of the corps main effor*t), and an indication of the top-level task to be used to accomplish that goal (e.g., *attack to seize OBJ SLAM*).

A COA *problem* is defined as a subclass of planning problem. The problem goal is the top-level goal indicated in the mission statement, and the rest of the mission statement is represented as a constraint on how to select tasks in the plan. The five task categories are represented as subplans (they are not subtasks or subgoals but useful categories for grouping the unit's activities). Each sentence in the statement is turned into a plan-task as follows. There is a specification of *who* is doing the task, e.g., *a mechanized brigade*, which is represented as the agent of the plan-task. There is an indication of *what* is to be done, e.g., *attacks to fix enemy forces*, which is interpreted as a fix plan-task (where *fix* is a kind of task that is a subclass of the class attack). The *why* (or *purpose*), e.g., *to deny enemy forces the ability to interfere with the* C*OA's main effort,* can be a state-based ("enable P", "prevent P") or action-based ("protect another unit from enemy"). Therefore, the ontology defines the *purpose* of a COA task as a goal specification that can be either an effect or a capability of the plan-task. The *where*, e.g., *in the North*, is the location of the plan task. The *when* clause (e.g., *H-hour D-day*) is represented as a temporal commitment or as an ordering commitment if it is specified with respect to another task. Finally, the *main effort* and *supporting efforts* are defined as specializations of the subtask relation.

It is noteworthy that PLANET does not include entities commonly associated with the planning domain such as agents, resources, times and locations. But PLANET does include an ontology of evaluation criteria used for the COA critiquing problem.

Moreover, separate ontologies can be integrated within PLANET. For example, PLANET has been used with an ontology of Allen's time relations.

5.3 Knowledge representation

PLANET ontology is available in different representation languages such as Loom, KIF, and MELD (CycL language). Documentation about its alignment with IKB (Integrated Knowledge Base) during the HPKB program is available at http://www.isi.edu/isd/HPKB/planet/alignment.

Also, PLANET was integrated into CYC, which resulted in an extension to the CYC IKB.

5.4 Use of the ontology and extensions

PLANET has been used to represent plans in three different domains: air campaign planning (ACP), army courses of action (COA), and enemy workarounds to target damage (WA).

Although all three are military domains, the plans represented are of a radically different nature. In the COA and WA domains, the plans were built manually by users and needed to be represented as given, even though they contained potential flaws and serious errors. In the ACP domain, plans are hierarchically decomposed and have verb-based objectives. Information about causal links and task decomposition templates is not provided. The COA domain has plans with a hierarchical flavour that is not always explicitly represented in the plan. In the WA domain, plans were generated automatically by an AI planner and are not hierarchical, consisting of a set of partially ordered steps and causal information in terms of the enabling conditions and achieved effects of each step.

5.5 PLANET-COA application

The PLANET-COA application is a critiquing tool that finds flaws in manually developed COAs for army operations at the division level. A COA is specified by a user as a set of textual statements (*who* does *what*, *when*, *where*, and *why*), together with a sketch drawn over a map. The PLANET-COA ontology allows the user to represent the COA that results from joining both text and sketch, which is the input to the critiquing tool.

An ontology like PLANET can provide a shared plan representation for systems to communicate and exchange information about the plan, and can facilitate the creation of a common, overarching knowledge base for future integration of planning tools.

PLANET was designed to be reused as an independent component, for example, as a knowledge component within an ontology-based translation service for planning agents.

6. Discussion

In this section, we discuss some aspects of earlier work in order to have a global view of the models proposed according to different perspectives, and draw some conclusions with respect to our work. We consider the scope and content of the ontologies examined in previous studies, the development methodology described, the knowledge representation techniques used, and the reuse or merging of existing ontologies.

6.1 Ontology content examined in previous studies

The ontologies described in this document cover both general knowledge entities relevant to the planning domain and more specific military concepts that are of importance during COA analysis.

6.1.1 General planning concepts

Planning models presented in the first part of the document (e.g., SPAR, CPR) are intended to represent general domain-independent concepts that facilitate the exchange of information between planning systems. They draw on previous or parallel initiatives in non-military domains (PIF/Process, scheduling ontologies). They are designed for plan representation of a more general nature than PLANET, for example. They also cover plan execution. The concepts covered by these models include but are not limited to the description of:

- actions, activities
- plan, constraints
- objectives
- evaluation criteria
- time and space
- resources
- world, world state

Although the models are intended to represent plans for the military domain, some modelling differences occur. Action, plan and objective are represented as different objects in CPR. An action in one context could be seen as a plan in another or an objective in another. Possibly they should be the same class or derived from a common superclass (e.g., CPR Design).

Moreover, when describing properties of plan-tasks, planners typically represent tasks differently depending on their approach to reasoning about actions.

Also, there are different ways to view the domain, e.g., when considering the static aspect of the planning domain (what is a plan? what are the related objects?) and the dynamic aspect related to execution of plans.

6.1.2 Specific military concepts

Most COA ontologies reported here cover more specific military concepts that are specializations of general and domain-independent upper-ontologies (e.g., Disciple). In particular, military units and military equipment are specified. They can be easily built from different information sources, for example, data sources describing military equipment (e.g., *Janes*). These concepts are also present in military data models such as the NATO C2IEDM data model. But to be able to exploit these ontological concepts, the ontology has to be formally specified including axiomatization, so that reasoning modules (intelligent agents, knowledge-based systems) can draw inferences from them (e.g., reason about the capabilities of a particular item of equipment).

COA development is a complex process involving the analysis of many knowledge components. Therefore, care must be taken to identify the relevant concepts to be considered and incorporated into a COA ontology. For example, the centre of gravity concept was explored in the Disciple project for COG analysis during COA development, and involved many specialized concepts to be considered (e.g., political and economic factors).

It is noteworthy that the representation of some important military concepts is not reported in the literature, for example, the concept of rules of engagement is not even mentioned in the studies surveyed.

6.2 Development methodology

Whereas the literature about ontological engineering provides numerous proposals for ontology development methods, the projects described herein do not cover this aspect in depth. In some cases, ontologies were built by a team of AI experts and validated by SMEs (e.g., CPR, SPAR) using an incremental process. In HPKB/RKF-related projects, ontologies are built by SMEs (after importing relevant concepts from existing ontologies) by using appropriate interfaces facilitating the capture of military knowledge through a dialogue. SMEs can themselves enter domain knowledge in the system (e.g., objects, critiquing rules for Disciple) without assistance from AI specialists or knowledge engineers. One drawback of this is that they can make errors when formulating domain knowledge.

Several projects took an incremental approach, drawing on previously constructed ontologies in the domain (CPR from KRSL, or SPAR from CPR).

Usually, ontological engineering favours a mixed top-down and bottom-up approach. High-level concepts are identified and imported from existing upper-level ontologies as they constitute reusable components in the domain. Then, more specific concepts are modelled according to the problem of interest (e.g., task → military-task → attack task, → penetrate-task).

A COA ontology should capture concepts that will not vary from scenario to scenario. However, scenario-based ontology development facilitates the identification of concepts that seem to be relevant in different use cases, and the concepts that are exploited during the reasoning process and should be included in the ontology. The question is: which of these concepts are needed in the majority of scenarios and therefore should be part of the ontology. In other words, what should be the scope of the ontology. Human subject matter experts face the same problem; however, through experience and study, they have developed a framework of concepts that they quickly consider as they "Assemble Relevant Data".

Some studies address the problem of managing large ontologies. They propose ontology structuring with contexts (e.g., JFACC, CYC) to facilitate this task.

6.3 Knowledge representation formalisms

As mentioned in the introduction, domain ontologies range from controlled vocabularies to formal computational ontologies. The ontological models reported in this document cover the whole spectrum.

- Informal ontologies consist of a shared vocabulary whose meanings are expressed in natural language.
- Object-oriented models graphically represented using UML (e.g., CPR). Objects are specified in hierarchies where relationships between objects are usually generalization (subclass/subsumes) or aggregation (part-of/contains) relationships.
- Formal ontologies with axiomatization.

Formal computational ontologies also contain formal specification of concept properties or constraints (axiomatization). Knowledge representation formalisms used in the projects presented herein are based on formal languages such as description logics or first-order logics. The reasoning capabilities they provide (e.g., inferencing) are exploited within applications (e.g., Disciple).

6.4 Ontolology import, merging and alignment

The development of ontologies has been proven a complex and time-consuming task. Thus, reusing existing ontological models built in external environments has been seen as a potential solution to the knowledge acquisition bottleneck involved in ontology development.

However, ontology merging or alignment is difficult in particular when different representation formalisms are used, and requires translation mechanisms. Furthermore, it is also an issue with a single formalism because of modelling differences and ontological mismatches. Consequently, tools supporting the merging of ontologies are needed in this context: 1) to help solve ontological mismatches, and 2) to help implement translation mechanisms between different representation languages. Some ontology management environments propose semi-automatic ontology merging tools.

6.5 Relations to upper ontologies

An upper ontology is limited to concepts that are meta, generic, abstract or philosophical, and hence are general enough to address (at a high level) a broad range of domain areas. An upper ontology does not include concepts specific to particular domains, but it does provide a structure upon which ontologies can be constructed for specific domains (e.g., medicine, finance, engineering, etc.). Examples of existing upper ontologies are CYC, SUO (Standard Universal Ontology), and SUMO (Suggested Upper Merged Ontology). Below we show how some of these upper ontologies have been exploited to reuse general ontological components.

6.5.1 CYC

CYC [16] is a large knowledge base (KB) of commonsense knowledge that contains over 100,000 atomic terms axiomatized by a set of over one million handcrafted assertions. CYC language (CycL), a variant of predicate calculus, is used to represent assertions in CYC. The KB contains both abstract and general concepts that form CYC Upper-ontology and is divided into microtheories incorporating more specialized concepts (e.g., descriptions of military equipment).

Teknowledge and Cycorp have been contributors to the HPKB/RKF programs. In particular, they built a COA ontology in the CYC representation language and tested CYC engine capabilities to provide COA analysis and critiquing. As a result, some projects incorporated relevant CYC components into their initial ontology (e.g., Disciple). As well, some ontological components of relevant value were exported back into CYC within specialized microtheories (e.g., Disciple, PLANET).

6.5.2 SUMO

Suggested Upper Merged Ontology (SUMO), developed by Adam Pease and Ian Niles of Teknowledge Corporation, was originally created by selectively merging concepts from several established ontologies into one consistent ontology. The CPR documentation describes mapping between CPR and SUMO.

7. Conclusion

The objective of this study was to analyze ontological models reported in the literature in the planning domain as a prerequisite of building an ontology for COA analysis. This analysis provided a better understanding of the planning domain and the knowledge concepts involved in the process. The literature survey reported herein highlights the complexity of the military domain and the planning process. Building a COA ontology requires an analysis of military knowledge elements involved in the COA development process (process-oriented development approach). The ontology development should be supported by a set of representative scenarios (use cases) to facilitate the task.

Even if an ontology is designed to specify relevant concepts of particular interest, it is important to define its scope in order to prescribe the set of concepts to be specified. This will be possible when the requirements of the anticipated system are more precisely defined. Knowing which aspects of the COA that the system should consider will help determine the concepts to be defined and/or refined. For example, in the Disciple project, the Disciple-COG ontology added new elements for the COG analysis from the previously defined COA ontology.

Publications on the projects reported here and information found on the Internet gave us a high-level view of the content of the ontologies, the knowledge representation and the development approaches used. Unfortunately, the contents of most ontologies from the DARPA projects (HPKB, RKF) are only available to project members; consequently, we could not benefit fully from these models, as we could obtain only the partial descriptions given in the publications. However, we were able to glean the main military concepts that have to be considered. Furthermore, some of the reported projects deal with COA analysis for an army tactical context that is very different from our project context. Hence, their focus is on some concepts (environment, terrain, resources, constraints) that should be looked at in a very different manner in an air/joint operations context.

We are also analyzing the NATO C2IEDM data model with a view to mapping the relevant concepts in the planning/COA domain that are highlighted in this study as they relate to the elements of the NATO data model. C2IEDM provides a widely recognized foundation for the representation of military concepts required for the exchange of information between C2 information systems. However, the model can be considered only as an informal ontology, since most of the battlefield elements it represents are described informally in the voluminous documentation. For example, a plethora of types of actions are described in the documentation but they are not formalized. Thus, a thorough analysis of the model should provide relevant input for the COA ontology.

Finally, given the complexity of the task itself, the research environment should facilitate the dynamic enrichment of the ontology to facilitate the management of evolving ontologies.

References

1. ATCCIS Working paper 5-7 version 5.0, Overview of the NATO LC2IEDM, NATO unclassified, 18 mars 2002.
2. J. Blythe, Y. Gil, A problem-Solving Method for plan Evaluation and Critiquing, 1999, *Proceedings of the 12th Banff Workshop on Knowledge Acquisition, Modeling, and Management (KAW'99),* Banff, Alberta.
3. M. Boicu, G. Tecuci, M. Bowman, D. Marcu, S.W. Lee and K. Wright, A Problem-Oriented Approach to Ontology Creation and Maintenance, In Adam Farquhar and Kilian Stoffel, (eds). *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99) Workshop on "Ontology Management"*, July 18-19, Orlando, Florida, AAAI Press/MIT Press, Menlo Park, CA. 1999.
4. M. Boicu, Gheorghe Tecuci, Bogdan Stanescu, Gabriel C. Balan and Elena Popovici, Ontologies and the Knowledge Acquisition Bottleneck, in *Proceedings of IJCAI-2001 Workshop on Ontologies and Information Sharing*, pp. 9-18. Seattle, Washington, August 2001. AAAI Press, Menlo Park, California, 2001.
5. M. Bowman (2002), A Methodology for Modeling Expert Knowledge that Supports Teaching-Based Development of Agents", PhD Dissertation in Information Technology, School of Information Technology and Engineering, George Mason University, Fairfax Virginia.
6. M. Bowman, A. M. Lopez, Jr., G. Tecuci (2001) Ontology Development for Military Applications, *Proceedings of the Thirty-ninth Annual ACM Southeast Conference*, Athens, GA, pp. 112-117.
7. M. Bowman, Center of Gravity analysis: Preparing for Intelligent Agents, http://www.iwar.org.uk/sigint/resources/cog-intel/Bowman_M__01.pdf
8. Y. Gil, J. Blythe, PLANET: A Shareable and Reusable Ontology for Representing Plan, *Proceedings of the AAAI Workshop on Representational Issues for Real-world Planning Systems*, 2000.
9. H. Chalupsky, R. MacGregor, Ontologies, Knowledge Bases and Knowledge Management, AFRL-IF-TS-TR-2002-163, Final Technical Report, July 2002.
10. V. K. Chaudhri, A. Farquhar, R. Fikes, P. D. Karp, and J. J. Rice (1998). OKBC: A programmatic foundation for knowledge base interoperability. In *AAAI Proceedings of the Fifteenth National Conference on Artificial Intelligence,* Madison, Wisconsin, July 26-30.
11. P. Cohen, R. Schrag, E. Jones, A. Pease, A. Lin, B. Starr, D. Gunning, M. Burke (1998), The DARPA High-Performance Knowledge Bases Project, *AI Magazine*, Winter 1998.
12. Core Plan Representation (CPR). http://reliant.teknowledge.com/CPR2/
13. CPR Design, http://reliant.teknowledge.com/CPR2/Reports/CPR-RFC4/Design.html

14. T. Gruber (1993), A translation Approach to Portable Ontology Specifications, *Knowledge Acquisition*, 5(2), pp. 199-220.

15. IET, Rapid Knowledge Formation Program, http://www.iet.com/Projects/RKF/

16. D. Lenat, (1995) CYC: A Large-Scale Investment in Knowledge Infrastructure, Communications of the ACM, November 1995, Vol. 38, No. 11.

17. D. L. McGuinness, (2002) Ontologies Come of Age, In Dieter Fensel, Jim Hendler, Henry Lieberman, and Wolfgang Wahlster, editors. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press.

18. Canadian Forces Operations - B-GG-005-004/AF-000.

19. A. Pease (1998). The Warplan: A Method Independent Plan Schema, in Proceedings of AIPS'98, Workshop on Knowledge Engineering and Knowledge Acquisition for Planning.

20. S. Polyak, A. Tate (1999) Planning Intiative: Shared planning and activity Representation – SPAR Version 0.2: Request for Comments. http://www.aiai.ed.ac.uk/~arpi/spar/spar-doc-02.html

21. A. Tate (1998), Roots of SPAR - Shared Planning and Activity Representation - *The Knowledge Engineering Review,* Vol 13(1), Special Issue on Ontologies (eds. Uschold. M. and Tate, A.), Cambridge University Press.

22. 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.

23. A. Tate (1996), Towards a Plan Ontology, AI-IA Newsletter (Quarterly Publication of the Associazione Italiana per l'Intelligenza Artificiale), Special Issue on "Aspects of Planning Research", Vol. 9 No. 1, pp.19-26.

24. A. Tate, B. Drabble, J. Dalton (1996), O-Plan: a Knowledge-Based Planner and its Application to Logistics, In Advanced Planning Technology, ed. A. Tate, AAAI Press.

25. M. Uschold, M. Gruninger (1996), Ontologies : principles, methods and applications, *Knowledge Engineering Review*, 11(2), , pp. 93-155.

26. A. Valente, Y. Gil, W. Swartout. "INSPECT: A Tool to Evaluate Air Campaign Plans". Internal Project Report, http://www.isi.edu/expect/link/papers-planning.html

27. 1 Canadian Air Division (1 CAD)/Canadian NORAD Region (CANR) Planning Guide

28. S. Polyak, A. Tate (1999) Planning Intiative Shared planning and activity Representation – SPAR : Request for Comments, version 0.2, http://www.aiai.ed.ac.uk/~arpi/spar/spar-doc-02.html.

29. G. Tecuci, M. Boicu, D. Marcu, B. Stanescu, C. Boicu, J. Comello, A. Lopez, J. Donlon and W. Cleckner (2002), "Development and Deployment of a Disciple Agent for Center of Gravity Analysis", In Proceedings of *the Fourteenth*

Innovative Applications of Artificial Intelligence Conference (IAAI-2002), July 28 - August 1, Edmonton, Alberta, Canada.

List of symbols/abbreviations/acronyms/initialisms

ACPT	Air Campaign Planning Tool
AI	Artificial Intelligence
ARPI	Darpa Rome Lab Planning Initiative
CFOPP	Canadian Forces Operational Planning Process
COA	Course of Action
COG	Center of Gravity
CPR	Core Plan Representation
DARPA	Defense Advanced Research Projects Agency
HPKB	High Performance Knowledge Base
<I-N-OVA>	Issues - Nodes - Orderings/Variables/Auxiliary (a constraint model of activity)
JFACC	Joint Force Air Component Commander
KB	Knowledge Base
KRSL	Knowledge Representation Source Language
OMWG	Object Model Working Group
RKF	Rapid Knowledge Formation
SPAR	Shared Planning and Activity Representation
SUMO	Suggested Upper Merged Ontology
SUO	Standard Universal Ontology

Internal Distribution list

1 - Director General

3 - Document Library

1 - Head/DSS

1 - Head/IKM

1 - Head/SOS

1 - A.-C. Boury-Brisset (author)

1 - M. Allouche

1 - M. Blanchette

1 - M. Bélanger

1 - J. Berger

1 - A. Boukhtouta

1 - A. Benaskeur

1 - R. Breton

1 - C. Daigle

1 - A. Guitouni

1 - H. Irandoust

1 - A.L. Jousselme

1 - P. Maupin

1 - S. Paradis

1 - F. Rhéaume

1 - J.M.J. Roy

1 - R. Rousseau

1 - P. Vallin

1 - A. Auger

1 - LCdr E. Tremblay

1 - LCdr E. Woodliffe

1 - Maj B. Deschênes

External Distribution list

1 - Director Research and Development Knowledge and Information Management (PDF file)

2 - Defence Research and Development Canada – Toronto

Attn: J. Baranski

D. Smith

1 - Director Science and Technology Air

1 - Director Science and Technology C4ISR

1 - Director Science and Technology Air 5 (DSTA 5)

1 - Director Aerospace Requirements 4

1 - Director Aerospace Requirements 4-2

1 - Director Science and Technology Air

Attn: Maj. Neil Knapp

1 - Canadian Forces Command and Staff College

Toronto

Attn: Commanding Officer

1 - 1 Canadian Air Division/Canadian NORAD Region

Winnipeg

Attn: Charles Hunter

1 - Département d’informatique

Pavillon Adrien Pouliot

Université Laval, Québec, G1K 7P4

Attn: Dr Bernard Moulin

FICHE DE CONTRÔLE DU DOCUMENT

1. PROVENANCE (le nom et l'adresse) Defence R&D Canada - Valcartier 2459 Pie-XI North Québec, QC G3J 1X8	2. COTE DE SÉCURITÉ (y compris les notices d'avertissement, s'il y a lieu) Unclassified

3. TITRE (Indiquer la cote de sécurité au moyen de l'abréviation (S, C, R ou U) mise entre parenthèses, immédiatement après le titre.)
Ontological models for military courses of action (U)

4. AUTEURS (Nom de famille, prénom et initiales. Indiquer les grades militaires, ex.: Bleau, Maj. Louis E.)
Boury-Brisset, A.-C., Champagne, C.

5. DATE DE PUBLICATION DU DOCUMENT (mois et année) September 2006	6a. NOMBRE DE PAGES 47	6b. NOMBRE DE REFERENCES 29

7. DESCRIPTION DU DOCUMENT (La catégorie du document, par exemple rapport, note technique ou mémorandum. Indiquer les dates lorsque le rapport couvre une période définie.)
Technical Memorandum

8. PARRAIN (le nom et l'adresse)

9a. NUMÉRO DU PROJET OU DE LA SUBVENTION (Spécifier si c'est un projet ou une subvention) 13DM22	9b. NUMÉRO DE CONTRAT
10a. NUMÉRO DU DOCUMENT DE L'ORGANISME EXPÉDITEUR TM 2003-320	10b. AUTRES NUMÉROS DU DOCUMENT N/A

11. ACCÈS AU DOCUMENT (Toutes les restrictions concernant une diffusion plus ample du document, autres que celles inhérentes à la cote de sécurité.)

- [x] Diffusion illimitée
- [ ] Diffusion limitée aux entrepreneurs des pays suivants (spécifier)
- [ ] Diffusion limitée aux entrepreneurs canadiens (avec une justification)
- [ ] Diffusion limitée aux organismes gouvernementaux (avec une justification)
- [ ] Diffusion limitée aux ministères de la Défense
- [ ] Autres

12. ANNONCE DU DOCUMENT (Toutes les restrictions à l'annonce bibliographique de ce document. Cela correspond, en principe, aux données d'accès au document (11). Lorsqu'une diffusion supplémentaire (à d'autres organismes que ceux précisés à la case 11) est possible, on pourra élargir le cercle de diffusion de l'annonce.)

13. SOMMAIRE (Un résumé clair et concis du document. Les renseignements peuvent aussi figurer ailleurs dans le document. Il est souhaitable que le sommaire des documents classifiés soit non classifié. Il faut inscrire au commencement de chaque paragraphe du sommaire la cote de sécurité applicable aux renseignements qui s'y trouvent, à moins que le document lui-même soit non classifié. Se servir des lettres suivantes: (S), (C), (R) ou (U). Il n'est pas nécessaire de fournir ici des sommaires dans les deux langues officielles à moins que le document soit bilingue.)

The mandate of the Technology Investment Fund Project entitled CoA Critiquing System for the Improvement of the Military Estimate Process is to investigate new concepts for building intelligent systems dedicated to the analysis and critiquing of military courses of action. Providing commanders with advanced decision aids requires thorough understanding of the processes involved, their information requirements, and the development of formal domain models (ontologies) upon which reasoning processes can be based (e.g., knowledge-based systems, intelligent agents). The document presents planning and courses of action models reported in the literature. For each model we identify the constituent elements and give relevant aspects of ontology development, such as knowledge representation techniques, methodology of development used, and exploitation of the ontology developed. The objective is to compile enough information about the different models to help us construct a COA ontology tailored to our requirements.

14. MOTS-CLÉS, DESCRIPTEURS OU RENSEIGNEMENTS SPÉCIAUX (Expressions ou mots significatifs du point de vue technique, qui caractérisent un document et peuvent aider à le cataloguer. Il faut choisir des termes qui n'exigent pas de cote de sécurité. Des renseignements tels que le modèle de l'équipement, la marque de fabrique, le nom de code du projet militaire, la situation géographique, peuvent servir de mots-clés. Si possible, on doit choisir des mots-clés d'un thésaurus, par exemple le "Thesaurus of Engineering and Scientific Terms (TESTS)". Nommer ce thésaurus. Si l'on ne peut pas trouver de termes non classifiés, il faut indiquer la classification de chaque terme comme on le fait avec le titre.)

Military planning, knowledge representation, ontologies

Defence R&D Canada

Canada's Leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE R&D DÉFENSE

WWW.drdc-rddc.gc.ca