From Visual to Logical Representation
A GIS-Based Sketching Tool for Reasoning about Plans

John Li, Cleo Condoravdi, Adam Pease

[jli, ccondora, apease]@teknowledge.com

Teknowledge Inc.

Abstract

Abstract: Multi-modal and heterogeneous logic reasoning is of increasing importance within the AI community. The GIS based ArcView COA Sketcher (ArCS) sketch and translation tool developed under DARPA's High Performance Knowledge Bases program is an example of an enabling tool towards that goal. Army Course of Action (COA) sketches can be drawn and translated automatically into statements in a formal logic with the tool. The statements are inputs to a geographic reasoner as well as systems reasoning about plans. This paper discusses the design of the sketching tool, and issues in creating an effective correspondence between the visual and logical representations of a COA.

Introduction

Multi-modal and heterogeneous logic reasoning is of increasing importance within the AI community. Visual presentation of information through diagrams, sketches and charts is so ubiquitous in human communication that it has long been desired to have automated reasoning systems taking visual representations as inputs. More generally, research effort has been devoted to developing logics for diagrammatic reasoning. For example, [Fisler, 1996] has developed a heterogeneous logic for hardware verification of design diagrams. A visual representation has also been used in the teaching of classical logic itself [Barwise & Etchemendy, 1995].

DARPA's High Performance Knowledge Bases (HPKB) program provided us with an excellent opportunity for developing tools for visual inputs to systems performing knowledge-rich symbolic reasoning. The purpose of the HPKB program is to advance the state of the art in knowledge representation and reasoning and create applications incorporating advanced AI techniques that are relevant to the military [Cohen et al 1998]. One application of this program is to develop systems for evaluating and critiquing operational army plans for courses of action on the battlefield.

Army Courses of Action (COAs) are high-level battle plans. They describe the intended actions of friendly troops on the battlefield on the basis of possible enemy deployment and actions. Human planners specify COAs by means of a sketch and a textual description in accordance with army practice [Army, 1997].

The sketch consists of standardized symbology placed on a map. It presents map-based information about the battlefield — the location of friendly and enemy troops, terrain features and obstacles, military regions and boundaries, possible battle positions and maneuver paths — as well as the planned tactical actions.

The textual description describes in a controlled English grammar the actions of the troops, their temporal relation and their intended purpose within the overall plan.

The COA statement and the COA sketch are in part complementary to each other but there is also significant overlap between them. As a result, the two parts of the COA need to be combined through a fusion process.

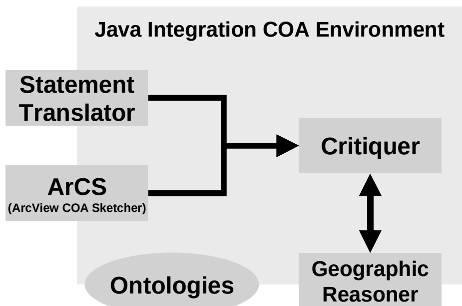
COA Sketches as Inputs to Plan Critiquers

The process of getting from the original COA inputs (statement and sketch) to the formal inputs needed for the critiquing systems presents challenges that are worth some discussion. One is that reasoning for plan evaluation relies on qualitative concepts and relative spatial relations between objects rather than the absolute geographical position of objects or the geographical profile of a region. However, it is the latter that is explicitly recorded in the sketch. For example, a critique checking to determine whether an action can successfully accomplish its stated purpose to enable another action will check whether the targets of the enabling action include all enemy troops on the path of the enabled action. This means that the reasoning system needs information making reference to concepts like spatial subsumption, betweenness along a non-cyclic path, trafficability of terrain, etc.

This implies that a translator should pass information about the absolute location of objects on to a geographic reasoner that can calculate qualitative spatial relations and estimate the trafficability of regions. Moreover, many concepts that are important to critiquers, such as the forward edge or the rear area of the battlefield, although implicitly depicted in the sketch, do not correspond to any specific symbol and need both geographic reasoning and domain specific knowledge to be identified with the proper region on the map.

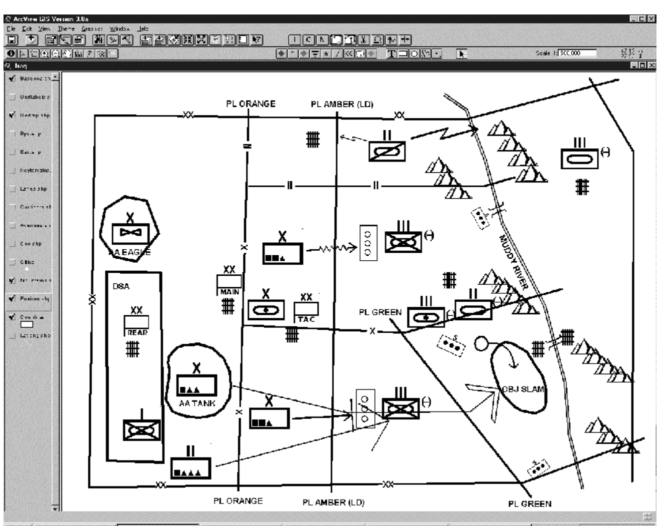
Our GIS based ArcView COA Sketcher (ArCS) sketch and translation tool was, therefore, designed to provide formal inputs to a geographic reasoner as well as systems reasoning about plans. COA sketches can be drawn and translated automatically into statements in a formal logic with the tool. In what follows, we will first discuss the software environment for COA critique and ArCS development. Then we will introduce an exemplar COA sketch drawn with ArCS and will use it in our discussion of the design of the translation process. Issues in creating an effective correspondence between the visual and logical representations of a COA are presented afterwards.

Execution Environment



We utilized Cyc [Lenat, 1995] as our knowledge representation and inference environment for the COA critique development. Relevant army doctrine is coded as logical statements in MELD/CycL [Cycorp, 1998]. Instances of COA statements and sketches are translated into MELD and imported into Cyc. Cyc is an inference system that can directly use these inputs to answer queries posed by the critiquer. Each COA critique was implemented as a *problem solving method* (PSM) [Gil, 1996] (this could be also termed a knowledge based procedure) and coded in Java. Each PSM made several calls to Cyc with a knowledge base query. Subsequent queries were created based on the results from earlier queries posed by the PSM. The most widely used PSM for COA critiquing could be called *critique and refine*. An initial query would be posed to determine if there was a problem of a particular sort. Further queries would zero in on the specifics of the problem so that the answer returned could provide the user with enough information necessary to effect a repair.

We utilized GeoRep [Ferguson, et al, 1999] as our geo-spatial reasoner. GeoRep can take as inputs the latitudes and longitudes of all objects drawn on the COA sketch and answer location and proximity questions with its qualitative spatial reasoning. It also provides trafficability support. For details see [Donlon et al, 1999].

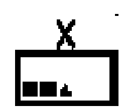


We used the ArcView Geographic Information System [ESRI, 1996:1] as our sketching environment. The Avenue [ESRI, 1996:2] scripting language was used both for handling user interaction and creating the MELD logical statement output created from the sketch. The sketch outputs in MELD followed the common ontology developed for the domain and were used by GeoRep, the Cyc-based critiquer and the critiquing systems of GMU (built with Disciple KA shell [Tecuci et al, 1999]) and ISI (built with the expect KA shell [Swartout & Gil, 1996]), and UMass's knowledge-based war gaming simulator [Atkin et al, 1999].

COA Example

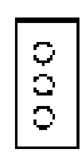
The graphic above shows a standard army COA drawn within ArCS. A human planner uses the ArCS tool to depict the deployment of friendly and enemy forces, the actions undertaken by friendly forces, the organization of the battlefield, including the areas of responsibility for friendly forces, and various features of the terrain. When the planner finishes the sketch, a click of the translation button allows thousands of logical statements to be generated instantly. This specific COA sketch, for instance, was translated into about 4,000 MELD statements.

To fulfill the drawing and translation function of ArCS, we made TrueType fonts for composable military symbols, and defined a mapping between the military symbology and ArcView's drawing functions, as well as another mapping between the military symbology and the MELD logical statements.

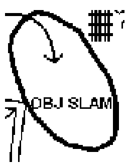


For example, the box symbol signifies a friendly military unit with the X denoting that the unit has the *echelon* or size of *brigade*. This particular unit is known as a *task organized composition unit*, which means that it is composed of subunits temporarily assigned for the purpose of a battle or larger operation. In this case there are two mechanized units (denoted by the squares) and one armored unit (denoted by the triangle) that comprise the brigade. The system knows that the next echelon smaller than brigade is *regiment* so it asserts that the subunits are regiment sized.

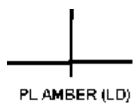
The arrows in the diagram denote actions that the units are supposed to undertake. The symbol denotes that the unit at the tail of the arrow is performing a *fix* task against the unit at the head of the arrow. A fix task is one in which the object is to prevent the enemy unit from moving. This can be done by artillery fire. In this particular case, the enemy is further constrained by the presence of a minefield (shown as three circles inside a vertical rectangle) between itself and the opposing unit.



The COA also contains named areas such as "Objective Slam" which serve as the targets of actions or zones of responsibility for particular units. The planner indicates that the friendly mission is to *seize* the objective by drawing the circle and arrow graphic pointing into the area.



Another important class of graphics is the lines that divide the battlefield. For example, the line, which has two vertical bars on it in the top center of the sketch, indicates that the area of responsibility for the *battalion* directly above it ends at the line. The vertical lines are called *phase lines* and provide an indication of the phasing of the plan. Phase lines indicate the position of units during different phases of the plan. The phase line acquires its full meaning when combined with temporal information provided in the COA statement. The vertical line which is marked "PL Amber" at the bottom center of the COA indicates that the line is named as indicated and is further specified as an "LD" or Line of Departure which denotes the point from which an attack takes place.



Translation of the Symbols

We first define a context or, in Cyc terminology, a *microtheory* in which all statements about the COA would be placed. This allows knowledge about the particular plan to be clearly differentiated from knowledge about the domain of all COAs. Also, by virtue of inserting the COA microtheory within the lattice of other Cyc microtheories, it allows for reasoning to be made more efficient by excluding inheritance from reasoning context which are irrelevant to the COA domain. In the MELD syntax below **`Constant`** introduces a new symbol to the system. **`in Mt`** indicates that all assertions below it are to be made within the given context. **`F`** indicates that a *formula* or logical statement is to follow. **`isa`** states that an instance is a member of a class. **`genlMt`** states that one microtheory inherits the contents of another.

```
Constant: BlueDivisionCOA1-1Mt.
in Mt: BaseKB.
F: (isa BlueDivisionCOA1-1Mt
        COASpecificationMicrotheory).
F: (genlMt BlueDivisionCOA1-1Mt
           ModernMilitaryTacticsMt).
```

The specification of the military units involved in the COA relies on the ontology of military units developed by [Andersen & Petersen, 1997]. Each unit has a specific military specialty, echelon, and strength. The relationInstanceExistsCount statements deserve further explanation. They are based on relations that are the MELD equivalents of frame predicates, which indicate that the relation given as the first argument holds between the instance given as the second argument and N instances of the class specified in the third argument where N is the fourth argument. This statement is equivalent to (following notation in [Sowa, 1999])

$(\exists x{:}\mathrm{arg3})(x@\mathrm{arg4} \wedge \mathrm{arg1}(\mathrm{arg2},x))$.

```
Constant: BlueMechBgd1.
in Mt: BlueDivisionCOA1-1Mt.
F: (isa BlueMechBgd1
       MechanizedInfantryUnit-MilitarySpecialty).
F: (isa BlueMechBgd1
        ArmoredUnit-MilitarySpecialty).
F: (echelonOfUnit BlueMechBgd1
                  Brigade-UnitDesignation).
F: (sovereignAllegianceOfOrg BlueMechBgd1
                             Blue-Side).
F: (troopStrengthOfUnit BlueMechBgd1
                        RegularStatus).
F: (relationInstanceExistsCount
       subOrgs-Direct
       BlueMechBgd1
       ArmoredUnit-MilitarySpecialty
       1).
F: (relationInstanceExistsCount
       subOrgs-Direct
       BlueMechBgd1
       MechanizedUnit-MilitarySpecialty
       2).
```

The system also names the areas that the units are placed in and have responsibility over. In the statements following, the shape and geographic location of areas of interest are given. While the Cyc-based reasoner does not use these metric points directly, they are generated and passed to the geographic reasoner which turns the metric statements into propositional, or relative statements about the geographic entities and passes those statements back to the critiquer.

```
Constant: Loc36.
in Mt: BlueDivisionCOA1-1Mt.
F: (isa Loc36 GeographicalRegion).
F: (objectFoundInLocation BlueMechBgd1 Loc36).
F: (shape Loc36 (AbstractFn Square)).

F: (longitude (CenterFn Loc36)
      (Degree-UnitOfAngularMeasure -97.5515)).
F: (latitude (CenterFn Loc36)
      (Degree-UnitOfAngularMeasure 37.8412)).

Constant: P241.
F: (isa P241 GeographicalThing).
F: (longitude P241
    (Degree-UnitOfAngularMeasure -97.5087)).
F: (latitude P241
      (Degree-UnitOfAngularMeasure 37.7984)).

;; etc (similar for other points)

F: (pointsOfBorder (BorderFn Loc36)
      (TheList P241 P242 P243 P244)).
```

Note that a lengthy specification of the coordinates specifying the center and the termini of the region (P241, P242, P243, and P244) is necessary for the geographic reasoner to recognize the shape of the region. These specifications for the regions, as well as those for the polylines and other geometric shapes, are the inputs to the geographic reasoner only. The critiquing systems use the derived results from the geographic reasoner such as the statements about the shape and degree of trafficability of a region given in the code below.

Because ArCS is built on top of a geographic information system, it is possible to overlay the sketch on digital terrain data and to use that information for further processing. One of the results of such processing [Donlon et al, 1999] is to generate information about the trafficability of regions. We can then attach to polygons that define those regions information about their trafficability.

```
Constant: COO2.
in Mt: BlueDivisionCOA1-1Mt.
F: (isa COO2 GeographicalRegion).
F: (degreeOfTrafficability COO2
        TerrainSeverelyRestricted).
F: (shape COO2 (AbstractFn Polygon)).
```

The sketch tool also provides the human planner with the ability to describe features that do not exist at the beginning of the plan but rather come into being as the plan unfolds, such as candidate battle positions. For example,

```
Constant: BP10.
in Mt: BlueDivisionCOA1-1Mt.
F: (isa BP10 GeographicalRegion).
F: (candidateBattlePositionOfCOA
       BlueDivisionCOA1-1Mt BP10).
F: (shape BP10 (AbstractFn Polygon)).
```

As discussed above, the phase lines and lines of responsibility collectively define regions that units are responsible for controlling at different points in time. Temporal information is not generated from the sketch and as a result not shown here. However, the process of translating the statements and combining them with the ArCS output does result in temporal information being provided.

```
Constant: Loc54.
in Mt: BlueDivisionCOA1-1Mt.
F: (isa Loc54 PhaseLineBoundedArea).
F: (sectorOfResponsibility BlueMechBgd1 Loc54).
F: (hasPartAsBorder Loc54 BoundaryBtnS).
```

The tasks assigned to units are also produced from the sketch and further specified in the text.

```
Constant: Fix1.
in Mt: BlueDivisionCOA1-1Mt.
F: (isa Fix1 Fix-MilitaryTask).
F: (unitAssignedToTask Fix1 BlueMechBgd1).
F: (objectActedOn Fix1 RedMechRegt1).
```

Issues in Translation

The representations provided above are relatively simple. The challenge was to define representations that could be easily and unambiguously generated by the sketch tool and yet still be powerful enough to represent all the information of interest and be merged successfully with the more complicated representations generated from parsing the controlled English COA text.

An additional complexity existed with regards to the domain. Because military COAs have been intended for understanding and use by humans, they contain considerable ambiguity and flexibility. For example, the presence of a unit at a particular point on a COA sketch does not necessarily mean that the unit will in fact be located at that precise point at any time during the battle. It is merely a candidate position that is consistent with the intent and responsibilities of the planner. The challenge was to provide the utility of such intended flexibility while not complicating the system design. In the end, this was handled by ensuring that the reasoning that operated on the sketch outputs was not sensitive to the exact positions. In this sense, the critiquing rules embodied a tolerance for imprecision. Accomplishing this required incorporating geographic reasoning which asserted qualitative statements about spatial relations as described earlier.

A similar consideration was that the units themselves are intended as prototypes or descriptions of units rather than references to real world units. For example, by placing a mechanized division symbol on the map, the human planner is not asserting that a particular division such as the 5th Mechanized division will be positioned at that location but rather that any available mechanized division can play that role in the plan. There again, the critiquing rules needed to take this into account.

Related Work

The nuSketch tool of Northwestern University [Forbus et al, 2000], under development during the HPKB program, has been intended to have a similar functionality to ArCS. It uses in addition a combination of speech and gesture to specify COA elements in order to free commanders from mouse and menus. Important differences include the fact that the ArCS system used a commercial geographic information system as its base.

Conclusion

The work described here was part of the effort to provide formal inputs to the various reasoning systems. We created a rich sketching tool that allowed a user to specify a battlefield plan in standard Army symbology. That symbology was converted to a logical representation, which combined with statement translation, could be effectively employed for reasoning about the suitability, feasibility and correctness of a military Course of Action.

Acknowledgements

We would like to acknowledge our DARPA sponsor, Murray Burke for supporting this effort. We would also like to acknowledge inspiration for, collaboration with and testing of ArCS from Ken Forbus and his team at Northwestern University. We would like to acknowledge Henry Gunthardt, SAIC, for his contribution in coding a drawing function.

References

Andersen, W. & Petersen, B. 1997, "Military Units Ontology", unpublished report.

Army field manual, FM 101-5. Headquarters, Department of the Army, Washington, DC, 31 May 1997.

Atkin, Marc, David L. Westbrook and Paul R. Cohen. 1999. Capture the Flag: Military Simulation Meets Computer Games. AAAI-99 Spring Symposium on AI and Computer Games.

Barwise, J. and Etchemendy, J. 1995, Hyperproof. Stanford CSLI Publications. Cambridge University Press.

Cohen, P., Robert Schrag, Eric Jones, Adam Pease, Albert Lin, Barbara Starr, David Gunning, and Murray Burke, 1998. The DARPA High Performance Knowledge Bases Project. AI Magazine, Winter, 1998. pp. 25-49.

Cycorp, 1998, "Features of the CycL Language", on-line report at http://www.cyc.com/cycl.html .

Donlon, J.J. and Forbus, K.D. 1999, Using a Geographic Information System for Qualitative Spatial Reasoning about Trafficability. Proceedings of the Qualitative Reasoning Workshop. Loch Awe, Scotland.

ESRI, 1996:1, "ArcView GIS: Using ArcView" ESRI software documentation, Redlands, CA.

ESRI, 1996:2, "Avenue: Using Avenue", ESRI software documentation, Redlands, CA.

Ferguson, R.W. and Forbus, K.D. 1999, GeoRep: A Flexible Tool for Spatial Representation of Line Drawings. Proceedings of the Qualitative Reasoning Workshop. Loch Awe, Scotland..

Fisler, K. 1996, A Unified Approach to Hardware Verification Through a Heterogeneous Logic of Design Diagrams. PhD Dissertation. Indiana University Department of Computer Science, August.

Forbus, K., Feguson, R., Usher, J., 2000, "Toward a Computational Model of Sketching", in preparation for AAAI-2000.

Gil, Y., and Melz, E., 1996, "Explicit Representations of Problem-Solving Strategies to Support Knowledge Acquisition". Proceedings of the Thirteen National Conference on Artificial Intelligence (AAAI-96), Portland, OR, August 4-8, 1996.

Lenat, D., 1995, "Cyc: A Large-Scale Investment in Knowledge Infrastructure." Communications of the ACM 38, no. 11, November.

Sowa, 1999, "Knowledge Representation: Logical, Philosophical and Computational Foundations", Brooks/Cole pub.

Swartout, W. and Gil, Y., 1996. "EXPECT: A User-Centered Environment for the Development and Adaptation of Knowledge-Based Planning Aids". In Advanced Planning Technology: Technological Achievements of the ARPA/Rome Laboratory Planning Initiative, ed. Austin Tate. Menlo Park, Calif.: AAAI Press.

Tecuci G., Boicu M., Wright K., Lee S.W., Marcu D., and Bowman M., 1999, "An Integrated Shell and Methodology for Rapid Development of Knowledge-Based Agents," in Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), July 18-22, Orlando, Florida, AAAI Press, Menlo Park, CA.