Course of Action Planning Ontology

Timothy P. Darr, Ph. D., Perakath Benjamin, Ph. D. and Richard Mayer, Ph.D.

Abstract—This paper describes an ontology to support course of action (COA) planning that provides an extensible framework for modeling COA plans consistent with Army and Marine Corp doctrine. This ontology is structured into a core ontology that includes definitions of common COA planning concepts (activities, phases, outcomes, measures-of-performance, measures-of-effectiveness, etc.), and multiple domain-specific ontologies that extend the core ontology for specific types of plans (stability operations, counterinsurgency planning, information operations planning, etc.). A preference relation between descriptions of the "plan state" using measures-of-effectiveness is introduced to allow subject-matter experts to specify a ranking over plan activities or phases from a specific Human Social Culture Behavior (HSCB) perspective. These preferences can be used in the planning process to identify or prune black holes and blind alleys.

Index Terms—Course of Action Planning, Information Operations, Counterinsurgency Planning, Utility Theory, Preferential Reasoning, Qualitative Reasoning.

I. Introduction

THIS paper presents the initial design of a Course-of-Action ontology (COA-Ontology) that can be used to support course of action (COA) planning. This ontology applies to the COA planning processes defined for the United States Army and Marine Corps for multiple domains, to include stability operations planning [1], counterinsurgency planning [2] and information operations planning [3]. The core ontology includes definitions of the common concepts and properties for defining COA plans, including: COAs, COA activities, COA phases, measures-of-performance (MOP) and measures-of-effectiveness (MOE).

The COA-Ontology consists of multiple sub-ontologies, each of which contains a small number of concepts and properties and can easily be integrated into other ontologies. For example, we have defined a measure-of-effectiveness (MOE) ontology that can be imported as a standalone ontology into other ontologies for the purpose of providing a common definition for representing MOEs. The urban Counterinsurgency (COIN) ontology is an example of how to extend the core ontology to define activities, MOP, MOE and phases that are tailored to a specific domain.

Qualitative MOPs and MOEs are used to describe plan states and outcomes. It is often the case that objectives are defined not in terms of fixed numeric quantities or ranges, but rather in qualitative terms such as: "reduce the number of attacks against coalition forces", or "increase the level of activity in the central market during daylight hours."

The Deep Maroon[1] ontology currently supports two types of reasoning: (i) utility-theoretic preferential or preference reasoning and (ii) goal-directed forward and backward chaining reasoning.

Preference reasoning provides a way to rank-order states from the perspective of a given community group (counterinsurgents, insurgent group, religious or ethnic group, etc.). An inference algorithm can use these preferences to reason about assessment of how a given activity will be perceived and can assist a planner in the identification of black holes or blind alleys[2].

Goal-directed forward and backward chaining reasoning provides a way to reason about the desired trajectory of the plan over time (forward chaining), and given an end state, determine a set of starting states that would result in that end state (backward chaining). In forward chaining, the sequence of activities that are available at each plan state can be determined by matching activity preconditions with the current state and asserting the new state that results from the application of the activity. In backward chaining, the possible states that can achieve a given outcome are determined, followed by the activities that can achieve that state. Interleaving forward and backward chaining with preference-based filtering helps to mitigate the complexity of developing and analyzing realistic plans[3].

A. Ontology Design Goals

The design of the COA-Ontology strives to achieve extensibility, flexibility and reuse. Our experience is that many ontologies are monolithic and cumbersome; making them difficult to understand and difficult to reuse. An example of what the authors believe is a concise, well-defined ontology

This work was supported by the Office of Naval Research under Contract N00014-09-C-0334.

Timothy P. Darr is a Research Scientist at Knowledge Based Systems Incorporated, College Station, TX 77840 USA (corresponding author to provide phone: 979-574-3189; e-mail: tdarr@kbsi.com).

Perakath Benjamin is Vice President for Research at Knowledge Based Systems Incorporated, College Station, TX 77840 USA (e-mail: pbenjamin@kbsi.com).

Richard Mayer is President of Knowledge Based Systems Incorporated, College Station, TX 77840 USA (e-mail: rmayer@kbsi.com).

[1] The name coined for this system by analogy to the DARPA Deep Green force-on-force planning and execution management initiative [7] and the IBM Deep Blue chess system.

[2] In the context of a COA plan, a black hole is a state that once you get into, you can never get out of. An example of a black hole is an activity that leads to an inflammatory situation such as civil war or increased intra-militia violence. A blind alley is a state that is unproductive in that there is no feasible next state or no path to a goal state. Unfortunately, in the COIN context blind alleys often turn into black holes as you may not be able to retrace to an earlier state.

[3] How the complexity is mitigated in this way is beyond the scope of this paper.

is the W3C geospatial ontology [4]. This ontology consists of only two concepts (`geo:SpatialThing`, `geo:Point`), and five properties (`geo:alt`, `geo:lat`, `geo:long`, `geo:lat_long`, `geo:location`): the minimal amount of ontological information to represent a geospatial region. From this ontology, more specific geospatial regions such as polygons, ellipses, and points can be defined by extending this geospatial ontology.

The COA-Ontology design principles are as follows[4]

- Ontologies should have a well defined purpose and support a well-defined set of use cases;
- Ontologies should include the minimal number of concepts and properties to support the purpose;
- Ontologies should strive to be more than a simple taxonomy of domain concepts; and
- When possible, allow for importing / exporting concepts and properties from / to other ontologies.

We realize that the design principles described above can be very subjective, but we believe that they are useful as a starting point for ontology design.

The purpose of the COA-Ontology is as follows:

- To represent COA plans that can be incorporated into other tools / applications, or to be used as a communication medium for the plans across systems.
- To represent concepts in the Human Social Culture Behavior (HSCB) modeling domain so that COAs may be assessed in a reusable, computer-process-able format from the perspective of multiple interested communities.

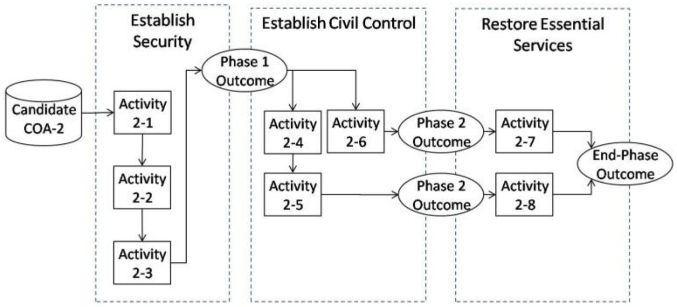
B. Paper Organization

The remainder of this paper is organized as follows. Section II defines the COA planning problem context. Section III describes the structure of the COA-Ontology family of ontologies. Section IV outlines the representation of measures-of-performance and measures-of-effectiveness. Section V describes the representation of COAs. Section VI describes the representation of preferences. Section VII describes inference support in COA-Ontology. Section VIII outlines conclusions and opportunities for future work.

II. Problem Context

This section describes the context that the COA-Ontology supports. Fig. 1 shows a simplified example of a stability operations COA plan. This figure shows one of three possible COAs that are proposed to achieve a commander's objective[5]. This COA consists of three phases: establish security, establish civil control and restore essential services. Each phase is terminated by an outcome that serves as a milestone for measuring progress of the plan. Each phase contains a sequence of activities that are performed to achieve the end-phase outcomes. The activities can be sequential, as shown in the establish security and restore essential services phases; or branch-and-sequence as shown in the establish civil control phase[6].

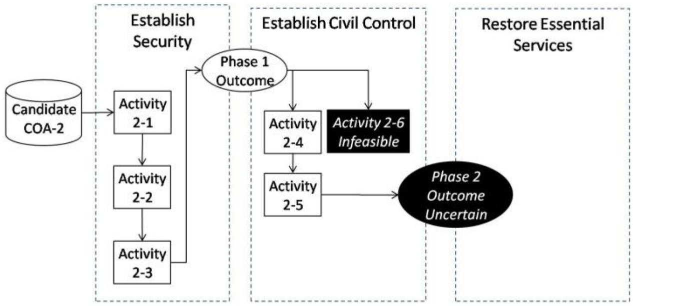
The forward chaining reasoning supported by the COA-Ontology can be used to reason from the initial state represented by the candidate COA on the left-hand side, to the activities that are possible at that state, to intermediate states that are achieved by each activity, to an end-phase outcome. The same reasoning is possible treating each end-phase outcome as an initial state. The backward chaining reasoning supported by the COA-Ontology can be used to determine what activities can achieve the end-phase outcome, backward through the states that enable the activities back to the initial state on the left-hand side.



Note: Phase names taken from Army FM 3-07 "Stability Operations"

Fig. 1 COA Planning Ontology Context

Fig. 2 illustrates the identification of black holes and blind alleys in a COA plan. Using the preference knowledge specified by an SME for a particular community segment, an inference engine can identify activities and outcomes as infeasible or uncertain, respectively. This has the potential to significantly reduce the search space as black holes and blind alleys are pruned from consideration or identified for further investigation.



Note: Phase names taken from Army FM 3-07 "Stability Operations"

Fig. 2 COA Planning Inference Support

III. Ontology Structure

This section describes the structure of the COA-Ontology. Fig. 3 shows the structure of the core ontology and an extension of the core ontology to support urban COIN COA planning. This organization allows users to import or use only those elements that are required for a given application, promoting flexibility and reuse.

[4] Similar to the principles defined in [8].

[5] Per doctrine, three COAs are typically presented to the commander for approval.

[6] It is also possible to have concurrent activities, though not shown in the figure.

The solid arrows represent ontology imports within the core and urban coin ontologies. The dotted arrows across the core / urban COIN boundary represents ontology imports from the urban COIN ontology to the core ontology.

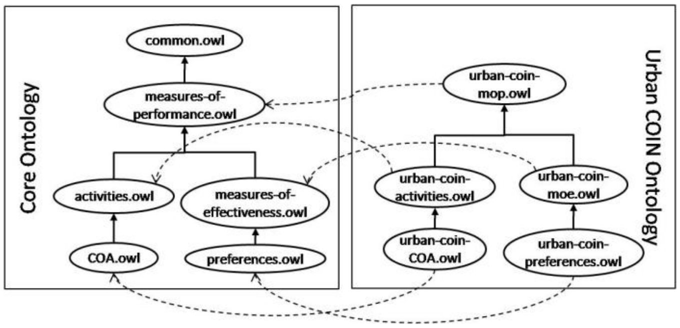


Fig. 3. HENIOMAP Core Ontology Structure

The core ontology consists of six sub-ontologies contained in six OWL files that define the most general concepts and properties. The core ontology is completely self-contained and can be used as the basis for more specific COA planning ontologies, promoting extensibility. The urban COIN ontology shown at the right of the figure is one such extension of the core ontology.

The common ontology (common.owl) contains definitions of general concepts and properties that are common to COA planning. The common ontology includes four classes:

- `IO-Thing` - a subclass of `owl:Thing` that is used to attach properties and relationships that are common to all classes in the ontology.
- `COA-Variable` - a subclass of `IO-Thing` that represents a variable that can describe some feature of a domain. A measure-of-performance or measure-of-effectiveness is a subclass of `COA-Variable`. This class has two properties: `hasValue` and `hasValueDirection`. The `hasValue` property is the actual value of the variable and can be one of the standard types (`xsd:integer`; `xsd:float`, etc.) as well as a qualitative value (see below).
- `Qualitative-Direction` - a subclass of `owl:Thing` that represents a qualitative description (increasing, decreasing, stable) of the direction or trajectory of an COA variable.
- `Qualitative-Values` - subclass of `owl:Thing` that represents qualitative values (HIGH, MEDIUM, LOW, etc.).

The measures-of-performance ontology (measures-of-performance.owl) contains the definition of measures of performance. According to COIN doctrine [2], a measure of performance is defined as "a criterion to assess friendly actions that is tied to measuring task accomplishment." MOPs in the COA-ontology are effectively state variables that are used to define the pre- and post-conditions for an activity and to be used as inputs to the calculation of MOEs. The MOP ontology includes a single class:

- `Measure-of-Performance` - a subclass of `COA-Variable` that represents a measure of performance. This class has the property `hasTimeStamp` to indicate the time at which the measurement was collected.

The measures-of-effectiveness ontology (measures-of-effectiveness.owl) contains the definition of measures of effectiveness. According to COIN doctrine [2], a measure of effectiveness is defined as "a criterion used to assess changes in system behavior, capability, or operational environment that is tied to measuring the attainment of an end state, achievement of an objective, or creation of an effect." MOEs in the COA-ontology define the commander's objective (end-of-COA outcome), objectives to be achieved at the end of each COA phase, and objectives to be achieved after each COA activity is applied at a given state. The MOEs depend on a set of MOPs. The MOE ontology includes two classes:

- `Measure-of-Effectiveness` - a subclass of `COA-Variable` that represents a measure-of-effectiveness. This class has the property `influencingMOP` that defines the set of MOPs that influence the MOE. An MOE can be views as a function that takes as input a set of MOPs and generates a measure. The `influencingMOP` defines the arguments to the function.
- `COA-Outcome` - a subclass of `IO-Thing` that represents an objective or outcome. This class has the property `hasMOE` that defines the MOEs that describe the outcome.

The activities ontology (activities.owl) contains the definition of a COA activity. The activities ontology includes a single class:

- `COA-Activity` - a subclass of `IO-Thing` that represents an activity within a COA phase. This class has four properties: `preconditionMOP`, `postconditionMOP`, `previousActivity`, `subsequentActivity`, and `hasActivityOutcome`. The `preconditionMOP` and `postconditionMOP` properties are used to define the precondition MOP for applying the activity and the state that results from applying the activity, respectively. The `previousActivity` and `subsequentActivity` properties are used to define a sequence of activities to perform within a COA phase. The `activityOutcome` property is the outcome that results from applying the activity.

The course-of-action ontology (COA.owl) contains the definition of a COA. The COA ontology includes two classes:

- `Course-of-Action` - a subclass of `IO-Thing` that represents a COA. This class has two properties: `hasPhases` and `hasOutcome`. The `hasPhases` property defines the phases within the COA. The `hasOutcome` property defines the commander's objective for the COA.
- `COA-Phase` - a subclass of `IO-Thing` that represents a COA phase. This class has four properties: `hasNextPhase`, `hasPrevPhase`, `hasActivities` and `hasOutcome`. The `hasNextPhase` and `hasPrevPhase` properties are used to define a sequence of phases within a COA. The `hasActivities` property defines the

activities within the COA phase. The `hasOutcome` property defines the outcome of the COA phase.

The preferences ontology (preferences.owl) contains the definition of preferences over COA outcomes. A preference in this context is a relation between two outcomes in which one of the outcomes is preferred to the other outcome, given the perspective of a specific human social culture behavior (HSCB) perspective. These preferences are typically asserted by an SME while role playing a specific HSCB perspective or community group. For example, in an agricultural community in which there is little or no electricity, a COA whose outcome involves restoration of economic self-sufficiency via the activity of building or restoring a canal system for crop irrigation, will be *preferred to* a COA in which the same outcome is achieved via the activity of providing electrical power to the local market. The preference ontology contains a single class:

- `Preference-Relation` - a subclass of `IO-Thing` that represents the pairwise preference between two outcomes from the perspective of a specific community group. This class has two properties: `lessPreferred` and `morePreferred`. The `lessPreferred` property refers to the outcome that is less preferred from the perspective of the community group. The `morePreferred` property refers to the outcome that is more preferred from the perspective of the community group.

KBSI is currently developing capabilities to reason about preferences of this kind using the application of Imprecisely Specified Multi-Attribute Utility Theory (ISMAUT) [5]. Preferences are used to model HSCB perspectives for the purpose of supporting the decision maker(s) and COA planner(s) in COA development, war gaming, comparison and decision making.

IV. Measures-of-Performance and Measures-of-Effectiveness

This section illustrates the relationship between MOPs and MOEs in more detail, as shown in Fig. 4. The namespaces for each of the ontologies are defined in the lower left of the diagram. An MOP has a unique timestamp and a value with a qualitative direction. An MOE is a specialization of an MOP with a set of MOPs that represent the arguments to a function that calculates the value of the MOE, given the values of the influencing MOPs. An outcome is described by one or more MOEs.

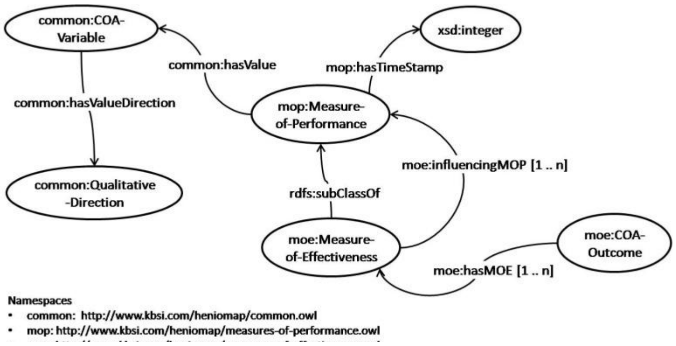

common:COA-Variable
common:hasValue
mop:hasTimeStamp
xsd:integer
common:hasValueDirection
mop:Measure-of-Performance
moe:influencingMOP [1 .. n]
common:Qualitative-Direction
rdfs:subClassOf
moe:Measure-of-Effectiveness
moe:COA-Outcome
moe:hasMOE [1 .. n]
Namespaces
• common: http://www.kbsi.com/heniomap/common.owl
• mop: http://www.kbsi.com/heniomap/measures-of-performance.owl

Fig. 4. Measures of Performance and Effectiveness

A. *Example - Urban COIN*

Fig. 5. shows example MOPs and MOEs from the COIN domain [2]. In this example, there is a single MOP, force density; two MOEs, establish presence and increase level of security; and a single outcome, increase level of security. Force density is a measure of the amount of force in a given area. The establish presence MOE is a qualitative measure of the amount of presence of U.S. and Host Nation (HN) forces in a given area. This could range from military patrols (U.S. only or U.S. and HN) or the establishment of police stations or outposts. The reduce reaction time is a statistical measure of the amount of time it takes to respond to a significant event in the area of interest (AOI). Each of these MOEs are a function of the force density, as well as other "state variables" (not shown) contained in an Intelligence Preparation of the Battlefield (IPB) such as the AOI, force structure and human terrain in the AOI. The 'increase level of security' outcome is a qualitative objective that measures whether or not there was an increase in the level of security in a given area. This objective is described by the MOEs, establish presence and reduce reaction time.

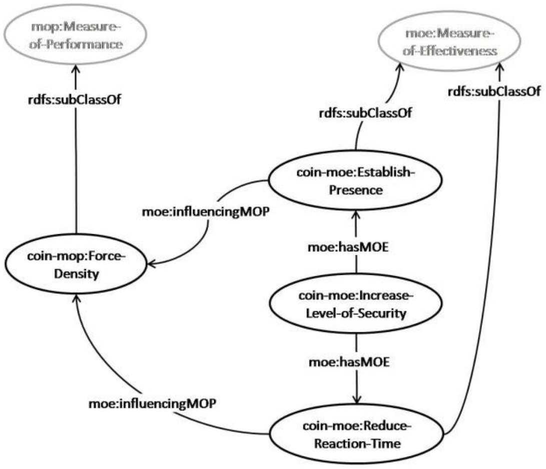

mop:Measure-of-Performance
moe:Measure-of-Effectiveness
rdfs:subClassOf
rdfs:subClassOf
rdfs:subClassOf
coin-moe:Establish-Presence
moe:influencingMOP
coin-mop:Force-Density
moe:hasMOE
coin-moe:Increase-Level-of-Security
moe:hasMOE
moe:influencingMOP
coin-moe:Reduce-Reaction-Time

Fig. 5. COIN MOPs and MOEs

V. Courses of Action

This section illustrates the relationship between COAs, COA phases, COA activities, outcomes and MOPs, as shown in Fig. 6. The namespaces for each of the ontologies are defined in the lower left of the figure. A COA consists of one or more phases. Each phase has an outcome and consists of a sequence of activities and may have a previous and next phase. Each activity has MOP pre- and post-conditions and an outcome. The precondition MOPs must be satisfied in order for the activity to be applied. Whenever the activity is performed, the postcondition MOPs are -determined.

A. *Example - Urban COIN*

Fig. 7 shows an example of a partial COA from the COIN domain [2]. Many of the details are missing, particularly the precondition and postcondition MOPs as described previously. In this example, there is one COA, two COA phases (one with an outcome shown) and three COA activities. The COA, labeled COA-1 has an establish security phase and an restore essential services phase, consistent with COIN and stability operations planning [1,2]. The establish security phase has an increase level of security outcome as described in the previous section. The establish security phase consists of activities for establishing access points, performing a census and establishing barriers. In this particular COA, the establish access points activity is succeeded by the perform census and establish barriers activities, which can be performed in parallel.

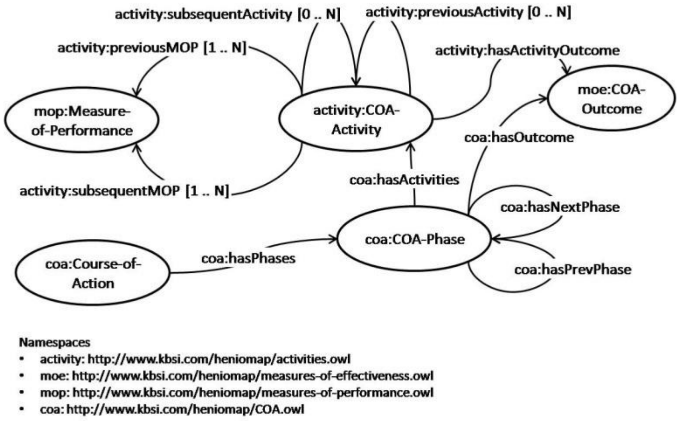

activity:subsequentActivity [0 .. N]
activity:previousActivity [0 .. N]
activity:previousMOP [1 .. N]
activity:hasActivityOutcome
mop:Measure-of-Performance
activity:COA-Activity
moe:COA-Outcome
coa:hasOutcome
activity:subsequentMOP [1 .. N]
coa:hasActivities
coa:hasNextPhase
coa:COA-Phase
coa:Course-of-Action
coa:hasPhases
coa:hasPrevPhase
Namespaces
activity: http://www.kbsi.com/heniomap/activities.owl
moe: http://www.kbsi.com/heniomap/measures-of-effectiveness.owl
mop: http://www.kbsi.com/heniomap/measures-of-performance.owl
coa: http://www.kbsi.com/heniomap/COA.owl

Fig. 6. Courses of Action

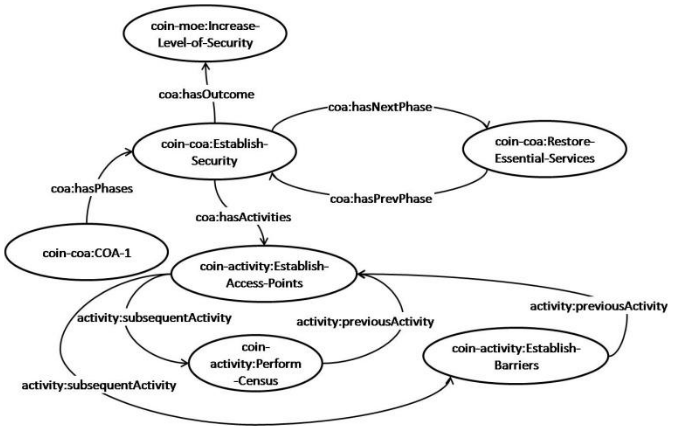

coin-moe:Increase-Level-of-Security
coa:hasOutcome
coa:hasNextPhase
coin-coa:Establish-Security
coin-coa:Restore-Essential-Services
coa:hasPhases
coa:hasPrevPhase
coa:hasActivities
coin-coa:COA-1
coin-activity:Establish-Access-Points
activity:previousActivity
activity:subsequentActivity
activity:previousActivity
coin-activity:Perform-Census
coin-activity:Establish-Barriers
activity:subsequentActivity

Fig. 7. COIN COA

VI. PREFERENCES

This section illustrates the relationship between preferences and outcomes, as shown in Fig. 8. A preference is a pairwise relationship between two outcomes, one of which is preferred to the other assuming a specific HSCB perspective.

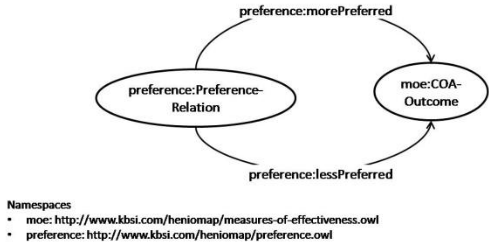

preference:morePreferred
preference:Preference-Relation
moe:COA-Outcome
preference:lessPreferred
Namespaces
moe: http://www.kbsi.com/heniomap/measures-of-effectiveness.owl
preference: http://www.kbsi.com/heniomap/preference.owl

Fig. 8. Preferences

VII. INFERENCE

This section describes the inference supported by the COA-ontology.

A. *Class Subsumption*

As much as possible, the COA-Ontology utilizes class subsumption via Description Logics (DL) based class definitions [6]. This section illustrates the COA ontology using an example from the COIN domain [2].

Fig. 9 shows a subsumption axiom for the MOP concept. This axiom states that every MOP is a concept such that there exists a `hasValue` relationship with an integer, float, boolean or qualitative value, and there exists a `hasValueDirection` relationship with a `Qualitative-Values` concept, and there is a `hasTimeStamp` relationship with an integer.

$$\begin{aligned}
&\forall x\ \exists y,z\ hasValue(x,y)\\
&\qquad \wedge \big(xsd{:}integer(y) \vee xsd{:}float(y)\\
&\qquad \vee xsd{:}boolean(y)\\
&\qquad \vee common{:}Qualitative-Values(y)\big)\\
&\qquad \wedge common{:}hasValueDirection(x,z)\\
&\qquad \wedge common{:}Qualitative-Values(z)\\
&\qquad \wedge hasTimeStamp(x)\\
&\qquad \rightarrow rdfs{:}subclassOf(x, Measure-of\\
&\qquad - Performance)
\end{aligned}$$

Fig. 9. MOP Subsumption Axiom

B. *Preferential Dominance*

The concept of preferential dominance is important to reasoning about preferences as it allows outcomes to be pruned very efficiently, thereby reducing the computational complexity, at very little cost, of searching through potentially very large outcome spaces. Intuitively, one outcome dominates another outcome if it is "better" than the other outcome along all the values of the variables that describe the outcomes.

Fig 10 shows the axiom for value dominance. The `atLeastAsGoodAs` relation is the greater-than-or-equal-to operator for numeric values. For qualitative values, this relation defined appropriately; for example, a "HIGH" value is "at least as good as" a MEDIUM value if the objective is to maximize the value of that variable.

$$\begin{aligned}\forall moe1, moe2, v1, v2\ (rdfs\colon subClassOf(moe1, Measure \\ - of - Effectiveness) \\ \wedge\ rdfs\colon subclassOf(moe2, Measure - of \\ - Efeectiveness) \wedge hasValue(moe1, v1) \\ \wedge\ hasValue(moe2, v2) \\ \wedge\ atLeastAsGoodAs(v1, v2) \\ \rightarrow valueDominates(moe1, moe2))\end{aligned}$$

Fig. 10. Value Dominance Axiom

Fig 11 shows the axiom for outcome dominance. Dominance is defined for MOEs first and then outcome dominance is derived by reasoning over the set of MOEs that describe each outcome. One outcome dominates another outcome if each MOE that describes the first outcome dominates the second outcome. Any outcome that is dominated by another outcome can be pruned from a search space, since it would never be chosen as there is a better or more preferred outcome.

$$\begin{aligned}\forall o1, o2, moe1, moe2\ (rdfs\colon subClassOf(o1, COA \\ - Outcome) \wedge rdfs\colon subClassOf(o2, COA \\ - Outcome) \wedge hasMoe(o1, moe1) \\ \wedge\ hasMoe(o2, moe2) \\ \wedge\ valueDominates(moe1, moe2) \\ \rightarrow outcomeDominates(o1, o2)\end{aligned}$$

Fig 11. Outcome Dominance

VIII. Conclusion and Future Work

This paper has described the preliminary design and of an ontology for describing COAs and some inference rules for using preferential reasoning to prune the space of possible outcomes. KBSI is in the process of implementing this ontology and using it to reason about COAs that have significant HSCB characteristics. More information about the COA ontology, including access, can be obtained by contacting the authors.

References

[1] "Stability Operations", Headquarters of the Department of the Army, Field Manual No. 3-07 (FM 3-07), October 2008.

[2] "Counterinsurgency", Headquarters Department of the Army, Field Manual No. 3-24 (FM 3-07), Headquarters Marine Corps Combat Development Command Department of the Navy, Marine Corps Warfighting Publication No. 3-33.5 (MCWP 3-33.5), December 2006.

[3] "Information Operations: Doctrine, Tactics, Techniques and Procedures", Headquarters Department of the Army, Field Manual No. 3-13 (FM 3-13), November 2003.

[4] http://www.w3.org/2003/01/geo/wgs84_pos

[5] White, C. C., *et al*, "A model of multi- attribute decision making and trade-off weight determination under uncertainty", IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-14, No. 2, pp. 223-229, 1984.

[6] The Description Logic Handbook: Theory, Implementation, and Applications, Baader, F. and Calvanese, D. and McGuinness, D. L. and Nardi, D. and Patel-Schneider P. F. (eds), Cambridge University Press, 2007.

[7] Surdu, J. R. and Kittka, K., The Deep Green concept. In *Proceedings of the 2008 Spring Simulation Multiconference* (Ottawa, Canada, April 14 - 17, 2008). Spring Simulation Multiconference. The Society for Computer Simulation International, San Diego, CA, 623-631. 2008.

[8] Gruber, T.R., "Toward principles for the design of ontologies used for knowledge sharing", *International Journal of Human-Computer Studies*, Vol. 43, Issues 4-5, November 1995, pp. 907-928.