Automated Site Selection of Air Defense Missile Batteries *

John Benton*
Modeling and Simulation Division
U. S. Army Topographic Engineering Center
Alexandria, VA 22315-3864.

V.S. Subrahmanian
Department of Computer Science &
Institute for Advanced Computer Studies
University of Maryland
College Park, Maryland 20742.

1 Introduction

In a war-time situation, Patriot and Hawk missile batteries are used to protect the front line against enemy attack. If the batteries could simply be positioned and left in place for the duration of the war, then the task of manually siting these batteries would not be onerous. Unfortunately, this is normally not the case during a shooting war. On a battlefront, these weapons systems may be moved several times in one week and there must be several contingency sites available for each battery. In particular, Hawk batteries have strict requirements for locating sites with optimum line-of-sight visibility, and even with current computer-assisted planning systems, siting the batteries is a trial-and-error based, time-consuming task.

Patriot Missile Batteries can be used to defend against either air-breathing threats such as jet aircraft or cruise missiles or against ballistic missile threats such as the Scud missile. Deployment patterns for the two tasks are quite different. In this paper, only the air-breathing case is considered.

The task of siting Patriot and Hawk missile batteries is tremendously simplified by recognizing the existence of several hierarchies in the problem domain. Patriot missile batteries are high cost systems which must be deployed sparingly. They are designed to attack high-altitude targets while they are still remote from the defended targets. These targets are typically high priority assets (command headquarters, ammunition dumps,etc). The Hawk missiles are lower cost systems and are intended to provide area defense and fill in the gaps left after placement of the Patriots. In turn, the Vulcans are positioned to provide close-in defense of the Hawks. As an example, if the Hawks are on a mountain ridge, their radars have a limited capability to look at angles below the horizontal. Vulcans on mountain sides below the Hawk emplacement can be used to defend the Hawks. There is a three-level hierarchy:

1. The Patriot missile batteries are deployed to protect high priority assets without consideration of where Hawk batteries will be sited,
2. The area defense provided by the Hawk batteries is intended to ensure that there are no gaps through which enemy craft can fly without opposition and
3. The Vulcans are positioned to protect the Hawk batteries.

There is a second hierarchy which can be defined in terms of the grid size used to represent the terrain. Given a 100 km by 100 km area that must be defended and terrain data at 100m spacing, the area will be represented by a 1000 x 1000 data array. The 100m spacing is good enough to find *plausible* areas for siting batteries. The detailed analysis of a specific site involves at least 100 rules and the application of several templates and also necessitates a higher resolution grid. A 30m grid spacing is minimally adequate for the detailed analysis of a site. In a hierarchical system, a Knowledge Based System (`KBS`) can be used to delineate plausible candidate areas and a local area `KBS` would be used to find sites that meet the detailed requirements for siting a battery. As an example, such an area

---

[*] This work was supported by the Army Research Office under Grant Nr. DAAL-03-92-G-0225 and by the National Science Foundation under Grant Nr. IRI-9357756.

might contain 4km^2 while each battery site requires approximate $0.1\ \text{km}^2$. The local area `KBS` could then return a list of satisfactory sites or return nil if no good sites are found. If the local `KBS` fails to find a site the global `KBS` presents an alternative area to the local `KBS`[1]. This paper will only analyze the global `KBS`.

2 A Knowledge Based Solution

Hybrid Knowledge Bases (`HKB`s) are a formalism proposed by Nerode and Subrahmanian that allow for the clean integration of multiple paradigms for *representing*, *reasoning*, and manipulating diverse forms of knowledge and data. Lu, Nerode, Remmel and Subrahmanian [7] have developed the mathematical foundations of (`HKB`s), while Subrahmanian [12] has set out the basic ideas behind `HKB`s. In this paper, we report on an ongoing interaction, which started in March 1992, that applies the theory and query processing algorithms associated with hybrid knowledge bases to the problem of missile deployment. In particular, we will show how the `HKB` framework may be used to naturally represent the problem of siting Patriot and Hawk missile batteries in order to protect assets located in a given theater of operations (`TOP`).

Deductive databases that provide database support for generating intelligent solutions to real-world problems must have the ability to deal with multiple modes of reasoning, including, but not limited to:

- reasoning about time,
- reasoning about quantitative relationships that may be expressed in the form of differential equations or optimization problems,
- reasoning about numeric modes of uncertainty about the domain which the database seeks to describe,
- reasoning about auxiliary data structures that may contain data of critical importance to the problem domain, and
- drawing “reasonable” conclusions based on assumptions of typicality and normality.

The problem of optimally siting a network of Hawk and Patriot missile batteries provides an exemplary problem domain where simultaneous reasoning about all aspects specified above are crucially necessary. Hybrid knowledge bases provide a sophisticated, and mathematically elegant, formalism to express all the above modes of reasoning, and hence, it provides a natural framework to articulate (and solve) the missile siting problem. In this paper, we will describe how `HKB`s can be used to solve the missile siting problem.

The paper is organized as follows. Section 3 describes the missile siting problem from an intuitive, mathematically clear, point of view. Section 4 informally describes the syntax and semantics of `HKB`s and shows how the missile siting problem can be solved using the `HKB` framework.

3 The Patriot/Hawk Missile Siting Problem: Mathematical Definition

A military planner who is allocating missiles to protect assets located in a given theater of operations must determine an appropriate location to site these missiles so as to afford maximal protection to the assets (in order to successfully withstand enemy attack). In general, the Patriot missile siting problem (`MSP`) involves:

1. A “theater of operations,” $\mathtt{TOP} = (\mathtt{RIGHT}, \mathtt{UP})$ describing the boundaries of a rectangular battlefield with x-coordinates stretching from 0 through `RIGHT` and y-coordinates stretching from 0 to `UP`, and
2. A finite set, `MISSILE_TYPE` of *missile types*, e.g. $\mathtt{MISSILE\_TYPE} = \{\mathtt{patriot}, \mathtt{hawk}, \mathtt{vulcan}\}$. and

---

[1] In order to simplify the problem, but without loss of generality, we assume that a given location (x, y) is sufficiently large to hold a missile battery. We know in actual fact that a battery site requires at least ten adjacent grid cells. An actual implementation of this problem will require the use of polygonally defined areas rather than the simple (x, y) coordinates used in this paper.

3. An availability function, $\psi_{\text{av}} : \texttt{MISSILE\_TYPE} \rightarrow \{0, 1, 2, \ldots\}$ reflecting the *availability* of different types of missiles, and

4. A set, `ASSETS`, of assets to be protected, and

5. A function $\phi_{\text{loc}} : \texttt{ASSETS} \rightarrow [0, \texttt{RIGHT}] \times [0, \texttt{UP}]$ which specifies the *location* of each asset, and

6. A function $\rho_{\text{prt}} : \texttt{ASSETS} \rightarrow \mathbf{R}$ which specifies the *priority* of an asset, and

7. A probability-of-kill template, which in this paper will be called an *effectiveness* function that is defined as follows:

   $$\epsilon_{\text{eff}} : \texttt{ASSETS} \times \texttt{MISSILE\_TYPE} \times [0, 360] \times [0, \texttt{RIGHT}] \times [0, \texttt{UP}] \times [0, \texttt{RIGHT}] \times [0, \texttt{UP}] \rightarrow \mathbf{R}$$

   which specifies how "effective" the placement of a specific missile at a specific location and a specific orientation[2] is vis-a-vis protecting an asset. Intuitively, $\epsilon_{\text{eff}}(a, mt, \theta, x_1, y_1, x_2, y_2)$ returns as output, a real number denoting the effectiveness of protecting asset a at location (x_1, y_1) by placing a missile of type mt at location (x_2, y_2) with orientation θ.) An example of such a function is

   $$\epsilon_{\text{eff}}(-, \texttt{patriot}, -, x_1, y_1, x_2, y_2) \quad = \quad \sin\left(\frac{\pi \times (\text{max\_range} - r)}{\text{max\_range}}\right) \times \cos\left(\frac{\pi}{2} \times \frac{\theta - \theta_{BS}}{\text{max\_angle}}\right)$$

   where $r \in [0, \text{max\_range}]$ is defined as the Pythagorean distance between the coordinates, (x_2, y_2), of the Patriot battery, and location of the asset to be defended, (x_1, y_1). θ_{BS} is the azimuthal angle of the radar bore-sight, max_range is the maximum offset of the Patriot from the asset it is defending and max_angle is the limiting azimuthal angle for which a Patriot site can defend an asset. The effectiveness function for the Hawk depends on the placements of the Patriot batteries.

8. A *net worth* function $\eta_{\text{nw}} : \mathbf{R} \times \mathbf{R} \rightarrow \mathbf{R}$ which specifies, given a priority and an effectiveness, the combined net worth of the two.

4 Using HKBs to Solve the Missile Siting Problem

In Section 3, we have formalized the problem of siting missile batteries so as to best protect a set of assets in a given theater of operations. We will now address the question of how the diverse databases relate to the HKBs and why they are a natural means to address (and solve) this problem.

The formalization of the missile siting problem given in the preceding section is a purely numerical one which does not address the following critical issues:

1. Terrain Features: Nowhere in the definition of MSPs given above are terrain features taken into account. Only geometric constraints are considered. This needs to be rectified (and will be rectified below).

2. Physical Constraints: It is not possible to site certain types of missiles at certain locations. For instance, Patriots cannot be sited in areas inaccessible by road, though a site that is inappropriate for siting Patriots may be appropriate for siting Hawks, and these factors must be taken into account. Thus, there are physical constraints that play a key role in identifying certain sites as being "inappropriate" sites for certain kinds of missiles. These physical requirements are expressed as constraints over the databases used to store terrain data.

---

[2] We assume that the orientation of an asset w.r.t. a missile is 0 degrees if the asset lies directly on the bore-sight of the Patriot missile radar, otherwise, the degrees are computed from 0 to 360 in the clockwise direction.

4.1 Terrain Databases

The main source of data are currently Digital Terrain Elevation Data (DTED) and Interim Terrain Data (ITD) from the Defense Mapping Agency (DMA). DTED data is simply a two dimensional array of elevation data provided on a CD-ROM. Most of this data is at a grid spacing of approximately 100 meters. In the future, increasing amounts of data will be available at 30 meter spacing. The ITD data consists of six types of map overlay data: (1) roads and transportation, (2) vegetation, (3) soils, (4) slope, (5) obstacles and (6) drainage. An additional product is the Condensed Army Map Mobility System (CAMMS) in which the maximum off-road velocity for a given vehicle is specified for each (x, y) coordinate of the map area. GIS query languages typically provide the capability to answer a question related to the characteristics of the point (x, y) or to return a list of polygonally defined areas that satisfy a query. Queries can be very complex involving boolean operations across an arbitrary number of the overlays. Some examples of GISs that may be involved are:

1. *Soil Overlays*: Soil overlays specify the soil type in any region of the `TOP`. Knowledge of soil type is important not only to mobility but also to load bearing calculations for specifying a missile site. In particular, we assume that there is a function `GIVE_SOIL`(x, y) that returns as output the soil classification of the point (x, y).

2. *Vegetation Overlays:* These are exactly like soil maps with the difference being that they contain vegetation information. We will assume that we have query functions that return whether a specific site is dry ground, swampy or a lake. For forested areas, we assume that we have functions that return information on average tree spacing and thickness of the trunks.

3. *Drainage Overlays:* For the sake of simplicity, we will assume that the database can be queried to determine if a specific site is dry ground, swampy or a lake. It is also important to know if the site is subject to flash flooding or if the area is likely to be water logged after a downpour. Queries involving these features can be used to determine if it is suitable for a specified activity.

4. *Road Overlays:* The network of roads is specified along with specific information on the type of road (paved, gravel, dirt, etc.) The GIS can also be queried for the location of a road nearest the specific site. However, the `TOP` may contain numerous other places that are not deemed worthy of serious protection. These places, however, may indeed be potential places where missiles could be sited.

In this paper, we will consider a GIS loaded with terrain data to be a Terrain Specific Database (TSD), potentially one of a collection, $DB_1, ..., DB_n$ of TSDs. In general, each TSD comes equipped with a set of operations (or queries) $OP(DB1i)$, $1 \leq i \leq n$, that the database is able to support. The GIS hides the internal data structure except insofar as what one can infer from the speed of certain queries and from the set of queries that are supported. Note that if required, an `HKB` can inter-operate with many different GIS's simultaneously. The software implementing the GISs may use internal data structures – these data structures, too, are hidden from the `HKB`. Thus, all that the `HKB` needs to know in order to inter-operate with a GIS is the set of operations supported by the GIS, and not with the internal implementation (within the GIS) of these operations, and/or the data structures which these operations manipulate.

4.2 Siting Patriot Missiles: Expressing Physical Constraints using `HKB`s

In this section, we express physical constraints that specify some conditions that a location must satisfy in order for it to be a suitable Patriot missile site. These conditions will access the diverse databases described in the preceding section, making use of some of the operations described above. Furthermore, uncertainty will play a key role in formulating these constraints.

Before explaining how `HKB`s can be used to solve terrain reasoning problems, we briefly explain the syntax and semantics of `HKB`s. An `HKB` is a collection of clauses of the form

$$\begin{aligned} A : [u_0, t_0] \quad \leftarrow \quad & \Xi_1 \,\&\, \ldots \,\&\, \Xi_k \parallel B_1 : [u_1, t_1] \,\&\, \ldots \,\&\, B_n : [u_n, t_n] \,\&\, \\ & \mathbf{not}(B_{n+1} : [u_{n+1}, t_{n+1}]) \,\&\, \ldots \,\&\, \mathbf{not}(B_{n+m} : [u_{n+m}, t_{n+m}]). \end{aligned}$$

where $A, B_1, \ldots, B_{n+m}$ are atomic formulas of standard logic, and $\Xi_1, \ldots, \Xi_k$ are constraints over domains $\Sigma_1, \ldots, \Sigma_k$, respectively. The u's are either constants (real numbers between $[0,1]$), or variables ranging over $[0,1]$, or complex evaluable function terms that evaluate to values in $[0,1]$. The t's are either sets of time points (non-negative integers), or variables ranging over sets of time points, or complex evaluable terms that evaluate to a set of time points. The above clause may be informally read as: "If each of the B_i's, $1 \leq i \leq n$, has certainty at least u_i at all time points in the set t_i, and, for each B_j, $(n+1) \leq j \leq (n+m)$, it is not provable that B_j has certainty at least u_j at all time points in the set t_j, and each of the Ξ_e's, $1 \leq e \leq k$, are solvable over their respective domains, Σ_e, then conclude that A has certainty at least u_0 at all time points in the set t_0." Examples of domains for Σ_e's include the real numbers with arbitrary linear and/or nonlinear functions and functionals, the domain of integers, the domain of quadtrees, the domain of binary trees having nodes possessing a certain structure, flat files, raster representations of maps, to name a few.

Example 1 The standard practice in siting Patriot missiles is to first identify (and hence eliminate) unsuitable sites, and then to review the remaining sites to determine which places will have maximal net worth. Below, we define a predicate called `unsuitable` to identify unsuitable sites, and later define suitable sites as those that cannot be proven to be `unsuitable`. This is an effective software engineering strategy because, as information gets more and more sophisticated, it may be possible to eliminate more and more sites by identifying them as being `unsuitable`. This may be accomplished by the addition of new rules, as and when they become available, defining the `unsuitable` predicate. Persons familiar with digital terrain data will note that for the sake of simplicity, we are using a simplified model that does not correspond exactly to the real ITD data.

Patriot missile batteries cannot be sited in locations that satisfy any of the following conditions:

- (Vegetation) If the location is densely forested, Patriots missiles cannot be sited. We will assume that an area is densely vegetated if the vegetation level of the location is over 40 percent
- (Soil Information) Patriots cannot be sited in places where the soil type is unsuitable (e.g. in the case of sand dunes).
- (Water Level) Patriots cannot be sited in places that have excessive water (e.g. lakes and swamps). In so far as Patriots are concerned, we will assume that 15 percent or more water content renders the place unsuitable.
- (Road Information) Finally, Patriots are extremely heavy batteries and hence cannot be airlifted to a given site. Consequently, only locations accessible by road are potential Patriot sites.

The above criteria can be used to define suitable sites for Patriot missiles as follows.

$$
\begin{aligned}
\texttt{unsuitable}(Loc, \texttt{patriot}) : [1, \mathbf{R}] &\leftarrow \texttt{at}(Loc, X, Y) : [1, \mathbf{R}]\ \& \texttt{unsuitable1}(\texttt{patriot}, X, Y) : [1, \mathbf{R}]. \\
\texttt{unsuitable1}(\texttt{patriot}, X, Y) : [1, \mathbf{R}] &\leftarrow \texttt{S} = \texttt{GIVE\_SOIL}(X, Y) \,\|\, \texttt{bad}(\texttt{S}, \texttt{patriot}) : [1, \mathbf{R}]. \\
\texttt{unsuitable1}(\texttt{patriot}, X, Y) : [1, \{T\}] &\leftarrow \texttt{WATER\_LEVEL}(\texttt{X}, \texttt{Y}, \texttt{T}) > 0.15 \,\|. \\
\texttt{unsuitable1}(\texttt{patriot}, X, Y) : [1, \mathbf{R}] &\leftarrow \texttt{VEG\_LEVEL}(\texttt{X}, \texttt{Y}, \texttt{forest}) > 0.40 \,\|. \\
\texttt{unsuitable1}(\texttt{patriot}, X, Y) : [1, \mathbf{R}] &\leftarrow \texttt{ACCESSIBLE}(\texttt{X}, \texttt{Y}) = \texttt{false} \,\|. \\
\texttt{suitable}(Loc, \texttt{patriot}) : [1, \mathbf{R}] &\leftarrow \texttt{at}(Loc, X, Y) : [1, \mathbf{R}]\ \& \mathbf{not}(\texttt{unsuitable1}(\texttt{patriot}, X, Y) : [1, \mathbf{R}]).
\end{aligned}
$$

Note the use of the various operations such as `VEG_LEVEL(X,Y,forest)`, `WATER_LEVEL(X,Y,T)` and `GIVE_SOIL(X,Y)` which access (i.e. query) the a priori terrain databases. Similar clauses can be used to express suitability conditions for Hawk and Vulcans. In the above, we assume that the predicate `bad(S,patriot)` has been defined suitably. The above clauses reflect usage of:

- Time: the third rule above shows variation in the predicate `unsuitable1` with the time annotation T.
- Usage of Auxiliary Data Structures: the data structures storing information about the terrain (soil maps, vegetation data, drainage data, etc.) are all queries above.

- Non-Monotonicity: The negation symbol, **not** reflects *nonmonotonic* negation. The last clause above shows the usage of nonmonotonic negation. It says that all sites are suitable locations for the Patriot if there is no proof that the location is unsuitable. □

Example 2 A *placement* of missiles of the specified `MISSILE_TYPE`s and in the specified numbers (i.e. as specified by the function ψ_{av}) is an assignment of a location to each missile. Each placement has an associated "goodness" – the higher the goodness, the better. We define below, a predicate, $\mathtt{placement}(\mathtt{L}, \mathtt{missile\_type})$ where `L` is a list of 5-tuples of the form: $\langle \mathtt{missile\_type}, \mathtt{missile\_number}, \mathtt{orientation}, \mathtt{xcoord}, \mathtt{ycoord} \rangle$. Thus, for instance, if the 5-tuple $(\mathtt{patriot}, 5, 125, 4, 6)$ is in `L`, then this says that "Patriot missile number 5 is located at point $(4, 6)$ with an orientation of 125 degrees." The *value of a placement* of missiles of a specified type is the sum of the net worth of that placement w.r.t. the assets. In other words, if a Patriot missile is at position (x, y) with orientation θ, then the value of placing this Patriot at this location is

$$\sigma_{\mathrm{val}}(\mathtt{patriot}, x, y) = \sum_{a_i \in \mathtt{ASSETS}} \eta_{\mathrm{nw}}(\rho_{\mathrm{prt}}(a_i), \epsilon_{\mathrm{eff}}(a_i, \mathtt{patriot}, \theta, x, y, x_i, y_i))$$

where (x_i, y_i) is the location of asset a_i. Thus, the net worth of placing a specific missile of type mt at location (x, y) is assessed by simply adding up the effectiveness of placing this missile at (x, y) for *each* asset in the theater of operations[3]. The total value of a placement, `L`, then, is

$$\sum_{(mt, mn, \theta, x, y) \in \mathtt{L}} \sigma_{\mathrm{val}}(mt, x, y),$$

i.e. we simply sum up the net worth of each missile placed in the theater of operations. This can now be encoded as the following two annotated clauses.

$$\begin{aligned} \mathtt{placement}([], \mathtt{MT}) : [0, \mathbf{R}] &\leftarrow \\ \mathtt{placement}([\mathtt{H}|\mathtt{T}], \mathtt{MT}) : [V + \sigma_{\mathrm{val}}(\mathtt{MT}, \mathtt{X}, \mathtt{Y}), \mathbf{R}] &\leftarrow \mathtt{H} = (\mathtt{MT}, \mathtt{MN}, \theta, \mathtt{X}, \mathtt{Y})\,\& \\ & \mathtt{suitable}(\mathtt{MT}, \mathtt{X}, \mathtt{Y}) : [1, \mathbf{R}]\,\& \mathtt{placement}(\mathtt{T}) : [V, \mathbf{R}]. \end{aligned}$$

□

Example 3 Example 2 specifies the conditions under which a placement of missiles (of a specified type) satisfies the physical constraints, and in this case, determines the net worth of that placement. Below, we define an *optimal placement* of the missiles of a specified type – one which maximizes net worth.

$$\mathtt{opt\_placement}(\mathtt{L}, \mathtt{patriot}) : [V, \mathbf{R}] \quad\leftarrow\quad V' > V \,\|\mathbf{not}(\mathtt{placement}(\mathtt{L}') : [V', \mathbf{R}]).$$

This clause says that `L` is the optimal placement (with net worth V) iff there does not exist a placement $\mathtt{L}'$ having net worth $V' > V$. □

Above, we have only written (some) clauses showing how information about siting of Patriot missile batteries can be encoded. Information about other types of missile batteries used for air defenses may be encoded similarly.

Finally, let us reconsider the problem of finding an optimal placement given two Patriot batteries and one Hawk missile battery. Military doctrine requires that we first place the Patriots and then the Hawks. That is, we must first find an optimal placement of the Patriots, using which, the Hawks can later be sited.

The request to find such an optimal placement can be expressed by the following query: $\mathtt{opt\_placement}(\mathtt{L}, \mathtt{patriot})$ where $L = \{(\mathtt{patriot}, 1, O_1, X_1, Y_1), (\mathtt{patriot}, 2, O_2, X_2, Y_2)\}$ The instantiation of the variables O_1, O_2 will specify the orientations of the 2 Patriots in the optimal configuration, while the (instantiated) points $(X_1, Y_1), (X_2, Y_2)$, will specify where the missiles are to be located. Note that it is quite possible that two or more missile batteries may be in close proximity. [4] Similar queries can be asked when we have different numbers and/or different types of missile batteries available to defend the theater of operations.

---

[3] In reality, instead of summation, a more complex aggregation operation may be used. This too can be expressed within the `HKB` framework by replacing the summation by a function performing the desired aggregation.

[4] The reason for having multiple batteries to protect an asset is that many aircraft may try to saturate the defense.

4.3 Siting Hawk Missiles

In the previous section, we have seen how Patriot missile batteries may be sited to protect a collection of assets. However, the siting of Patriots may leave holes and weak points in the defense of the front line, which may allow enemy aircraft to penetrate without opposition. Consequently, not only do we need to protect assets, but we also need to close the gaps left by the Patriot siting. This task usually is accomplished by Hawk batteries.

Using the `HKB` framework described in the previous section, we can compute a siting, $\mathcal{S}$, for Patriot batteries. Let $\mathcal{S} = (x_1, y_1, \theta_1), \ldots, (x_k, y_k, \theta_k)$ be the positions and orientations of the Patriots. Let `defend` be a defendability function which specifies how well a point in the field of operations is defended by the Patriot sitings. Mathematically, `defend` is a function that takes as input, a point (x, y) in the field of operations and a siting $\mathcal{S}$, and returns as output, a non-negative real number in the $[0, 1]$ interval reflecting the defendability of the point (x, y). This function may be implemented using suitable constraint solving and other numerical techniques.

Hawks are sited at locations (x_h, y_h) which "cover" or defend global minima of the `defend` function. This can be specified as the `HKB` clause:

$$\texttt{locate1}(\texttt{hawk}, X_h, Y_h, \theta_h) : [V, \mathbf{R}] \quad \leftarrow \quad (X', Y') = \underline{\mathsf{min}}\, \texttt{defend}(X, Y, \mathcal{S}) \,\|\, \texttt{covers}(X_h, Y_h, \theta_h, X', Y') : [V, \mathbf{R}].$$

Here, `covers` is a covering predicate which specifies how effective siting the Hawk at location (X_h, Y_h) is in defending the gap at location (X', Y'). The predicate `covers` may be implemented in a variety of ways using linear and/or nonlinear mathematical computations. The above predicate, `locate1`, does not take into account, the visibility properties of the Hawk missile site (X_h, Y_h). Not only do we want the Hawk site to cover the gaps in the defense geometrically, but we also want visibility to be optimized (as argued by [8, 10]). This is expressed by the predicate `locate` below.

$$\texttt{locate}(\texttt{hawk}, X_h, Y_h, \theta_h) : [V, \mathbf{R}] \quad \leftarrow \quad (X_h, Y_h) = \underline{\mathsf{max}}\, \texttt{visible}(X, Y) \,\|\, locate1(\texttt{hawk}, X, Y, \theta_h) : [V, \mathbf{R}].$$

Here, the function `visible` is again a numeric function defined using non-logic-based representations.

Thus, we see that the notion of "asset" changes from missile to missile. As far as Patriots are concerned, assets are physical entities (military or civilian establishments) located on the ground. As far as Hawks are concerned, from a theoretical point of view, assets are just gaps in the air defense provided by the Patriots. Finally, we observe that Hawks themselves may need protection, especially when they are situated on hilly terrain (e.g. on mountain ridges). Hawks are typically defended by Vulcans – thus for Vulcans, the assets being protected are Hawk missiles. Siting of Vulcans is accomplished in a similar way (though the details and computations vary widely) to that of Patriots or Hawks.

5 Conclusions

In this paper, we have shown how hybrid knowledge bases provide a clean, mathematically elegant, practically viable, scalable solution for the problem of siting missiles when diverse topographic and terrain data is used in making such decisions in an intelligent, optimal manner. We are currently working on implementing a full-fledged hybrid knowledge base system for solving more general versions of the missile siting problem.

Terrain reasoning provides an important domain wherein we may test various theories dealing with hybrid reasoning, and reasoning with diverse data structures. This has been known – for instance, Antony [1] presents data structures incorporating both spatial and object-oriented reasoning in the terrain reasoning domain. In a similar vein, Hayslip and Gilmore [4] address the problem of representing complex knowledge about low altitude air combat over hilly terrain in the context of Grumman's `REACT` system. Both Antony's solution [1] and Hayslip and Gilmore's [4] techniques develop engineering and systems issues related to the heterogeneous data integration problem. However, until the advent of Subrahmanian's theory of mediators [11] and Nerode and Subrahmanian's concept of hybrid knowledge bases [12, 7], there has been no uniform theoretical framework for integrating diverse forms of knowledge and data, especially when these data and knowledge representation schemes contain temporal information, uncertainty, non-monotonic methods for reasoning about normality and typicality, and require us to access, reason with, and manipulate diverse data structures.

Acknowledgements

We have benefited from conversations with Anil Nerode and Anne Werkheiser. We thank Dave Hislop for introducing the authors to each other and suggesting that the hybrid knowledge base theory and mediator framework would find rich applications in the terrain reasoning domain.

References

[1] R. Antony. (1988) *Representation Issues in the Design of a Spatial Database Management System*, in: (J. Benton, ed.), Proc. Army Symp. on AI Research for Exploitation of the Battlefield Environment, Nov. 1988.

[2] C. Bell, A. Nerode, R. Ng and V.S. Subrahmanian. (1991) *Computation and Implementation of Non-monotonic Deductive Databases*, submitted to a technical journal. Currently available as University of Maryland Technical Report CS-TR-2801.

[3] C. Bell, A. Nerode, R. Ng and V.S. Subrahmanian. (1991) *Implementing Deductive Databases by Linear Programming*, Proc. 1992 ACM SIGACT/SIGART/SIGMOD Symp. on Principles of Database Systems, pps 283–292. Full version available as University of Maryland Technical Report CS-TR-2747, Aug. 1991.

[4] I.C. Hayslip and J. F. Gilmore. (1988) *A Multi-Level Knowledge Representation for Reasoning About Terrain*, in: (J. Benton, ed.), Proc. Army Symp. on AI Research for Exploitation of the Battlefield Environment, Nov. 1988.

[5] M. Kifer and V.S. Subrahmanian. (1989) *Theory of Generalized Annotated Logic Programming and its Applications*, Proc. 1989 North American Conf. on Logic Programming, MIT Press.

[6] M. Kifer and V. S. Subrahmanian. (1992) *Theory of Generalized Annotated Logic Programming and its Applications*, Journal of Logic Programming, Vol. 12, 4, pps 335–368, 1992.

[7] J. Lu, A. Nerode, J. Remmel and V.S. Subrahmanian. (1993) *Towards a Theory of Hybrid Knowledge Bases*, Univ. of Maryland CS-TR-3037. Submitted for publication.

[8] C. Ray. (1993) *Determining Visibility for Siting Problems*, Ph.D. dissertation, in preparation. Rensselaer Polytechnic Institute.

[9] H. Samet. (1989) *The Design and Analysis of Spatial Data Structures*, Addison-Wesley.

[10] R. E. Shannon and J. P. Ignizio. (1971) *Minimum Altitude Visibility Diagram – MAVD*, Simulation, pp 256-260.

[11] V.S. Subrahmanian. (1992) *Amalgamating Knowledge Bases*, Univ. of Maryland CS-TR-2949, August 1992. Submitted to ACM Trans. on Database Systems, Aug. 1992. Revised May 1993.

[12] V.S. Subrahmanian. (1993) *Hybrid Knowledge Bases for Intelligent Reasoning Systems*, invited address at the 1993 Conference on Logic Programming (GULP-93), Gizzeria, Italy, June 15-18, 1993.

[13] J. D. Ullman. (1988) *Principles of Database and Knowledge-Base System, Vol. 1*, Computer Science Press.

[14] G. Wiederhold, S. Jajodia, and W. Litwin. (1991) *Dealing with granularity of time in temporal databases*, in: Proc. 3rd Nordic Conf. on Advanced Information Systems Engineering, Lecture Notes in Computer Science, Vol. 498, (R. Anderson et al. eds.), Springer-Verlag, 1991, pages 124–140.

[15] G Wiederhold, S. Jajodia, and W. Litwin. (1993) *Integrating temporal data in a heterogeneous environment*, in: “Temporal Databases,” Benjamin Cummings.