Database Security

Course 5

The inference problem (part 2)

Processing of security constraints for inference control (continued).
Conceptual structures for inference control.

Plan

1. Processing the security constraints during the database design
2. Processing the security control and the release control
3. Conceptual structures for inference control
   - 3.1 Semantic networks and the inference problem
   - 3.2 Multilevel Semantic Networks
   - 3.3 Reasoning with Multilevel Semantic Networks
   - 3.4 Conditional statements and auxiliary networks
4. Enforcing the security constraints

1. Processing the security constraints during the database design (1)

- An **association-based constraint** classifies a collection of attributes, taken together, at a certain level of security.
- Such a constraint can generate relationships between different attributes.

1. Processing the security constraints during the database design (2)

- *Example*:
  - Given the relation SHIP (S#, SNAME, CAPTAIN) and an association-based constraint that classifies the attributes **SNAME** and **CAPTAIN**, taken together, at the **Secret** level, then one of the pairs **(S#, SNAME), (S#, CAPTAIN)** should be, also, classified at the **Secret** level.
  - Otherwise, an Unclassified user can get the pairs (S#, SNAME) and (S#, CAPTAIN), and then he can deduce the secret association (SNAME, CAPTAIN).

1. Processing the security constraints during the database design (3)

- An algorithm that processes **a given set of association-based constraints** and determines the **schema** for the multilevel database has been designed .
- Given the set of association-based constraints and an initial schema, the algorithm will determine **the attribute groups (clusters)** and the security level of each cluster.

1. Processing the security constraints during the database design (4)

- Such an algorithm must not only be executed at the time of design but can also be used **during the queries processing**.
- The query processor can **examine** the attributes of the various clusters generated by the algorithm and then determine the information to be **delivered** to users.
- For example, if the algorithm places:
  - attributes A1, A2 in *cluster* 1 at level *L* and
  - attributes A3, A4 in *cluster* 2 at level *L*,

  then, after an attribute in *cluster* 1 has been delivered to a user at level *L*, an attribute in *cluster* 2 cannot be delivered to the users at level *L*.

1. Processing the security constraints during the database design (5)

- Because **simple constraints** can be seen as a particular case of **association-based** constraint, where only one attribute is classified, it appears that these constraints can also be treated at the time of database design.
- Another constraint that can be addressed when designing the database is the **logical constraint**.
  - For example, if an attribute A implies an attribute B, and attribute B is classified at the Secret level, then attribute A must be classified at least at the Secret level.

1. Processing the security constraints during the database design (6)

- If **a constraint has conditions** associated with it, then dealing with it at the time of design can become difficult.
- For example, consider the constraint: "SNAME and LOCATION taken together are at the Secret level if LOCATION belongs to region X". Such a constraint **depends on the values of the data**, so it is better handled while processing the query or update.

- The algorithm of constraint processing in the database design step works as described subsequently.

1. Processing the security constraints during the database design (7)

- ***Association-based constraints.***

- *Input*: The set of association-based constraints and a set of attributes.
- *Output*: The set of clusters for each security level.
- For a security level *L*, each *cluster* will have a collection of attributes that can be safely classified at the level *L*
- If *A*1, *A*2, and *A*3 are attributes in a cluster *C* at the Secret level, then attributes *A*1, *A*2, and *A*3 can be safely classified together at the Secret level without violating security.
- Once the *clusters* are formed, **the database can be defined according to the functional and multi-value dependencies** that are applied.

1. Processing the security constraints during the database design (8)

- ***Simple Constraints***

- If an attribute *A* in the relation *R* is classified at the level *L*, then all the elements belonging to *A* are also stored at the level *L*.
- Therefore, we can store *A* at the level *L*.
- The algorithm that deals with simple constraints results directly.
  - Each attribute that is classified by a simple constraint is stored at the level specified in the constraint.
  - After the algorithm for processing simple constraints has been applied and the corresponding schema is obtained, then this schema is given as an input to the algorithm for processing the association-based constraints.
  - These constraints are then applied, and the final schema is obtained.

1. Processing the security constraints during the database design (9)

■ ***Logical Constraints***

- These are **rules that can be used to deduce new data from existing data**.
- If a security constraint classifies new data at a higher level than existing data, then **existing data must be reclassified**.
- The logical constraints can be either direct (Ai => Aj) or complex (for example : A1 & A2 & A3 &…& An => Am).
  - If Aj is classified at the Secret level then Ai must be classified at least at the Secret level.
  - If Am is classified at the Secret level, then at least one of A1, A2,…, An must be classified at least at the Secret level.

2. Processing the security control and the release control (1)

- We have presented **an integrated architecture** for a centralized environment, mainly in which the constraints are examined at the time of query, update or design.
- This implies, in the case of the query operation, that a large part of the activity is performed **before** the query is sent to the MLS / DBMS. After the execution of the query, certain constraints related to the delivery of the result are examined.
- More recently, research has been conducted on constraint processing only after the response is provided by the DBMS, but before it is delivered to the user. The idea behind this research is that **queries cannot be changed in certain cases**, especially if there are many constraints.

2. Processing the security control and the release control (2)

- This approach has several disadvantages.
- After the system delivers the response, all the security constraints must be examined and the information to be provided to the user must be determined. Many of the operations performed by the DBMS will need to be performed by the **Release Control Manager (RCM)**.
- The advantage of the approach is that, if there are many constraints, the complex process of modifying (rewriting) the query is avoided.
- The DBMS will produce the result **at the user's security level**. **RCM** will analyze this **result** and the **constraints**, and then determine if all the data obtained can be **delivered**.

2. Processing the security control and the release control (3)

- We assume that there is a constraint by which the values of the **LOCATION** column in the MISSION table are **Secret** and the **user** level is **Unclassified**.
- The provided data will contain all the information from the MISSION relationship.
- The previous constraint will be applied by the RCM and will only deliver the allowed values.
- Figure 1 illustrates the RCM module.

2. Processing the security control and the release control (4)

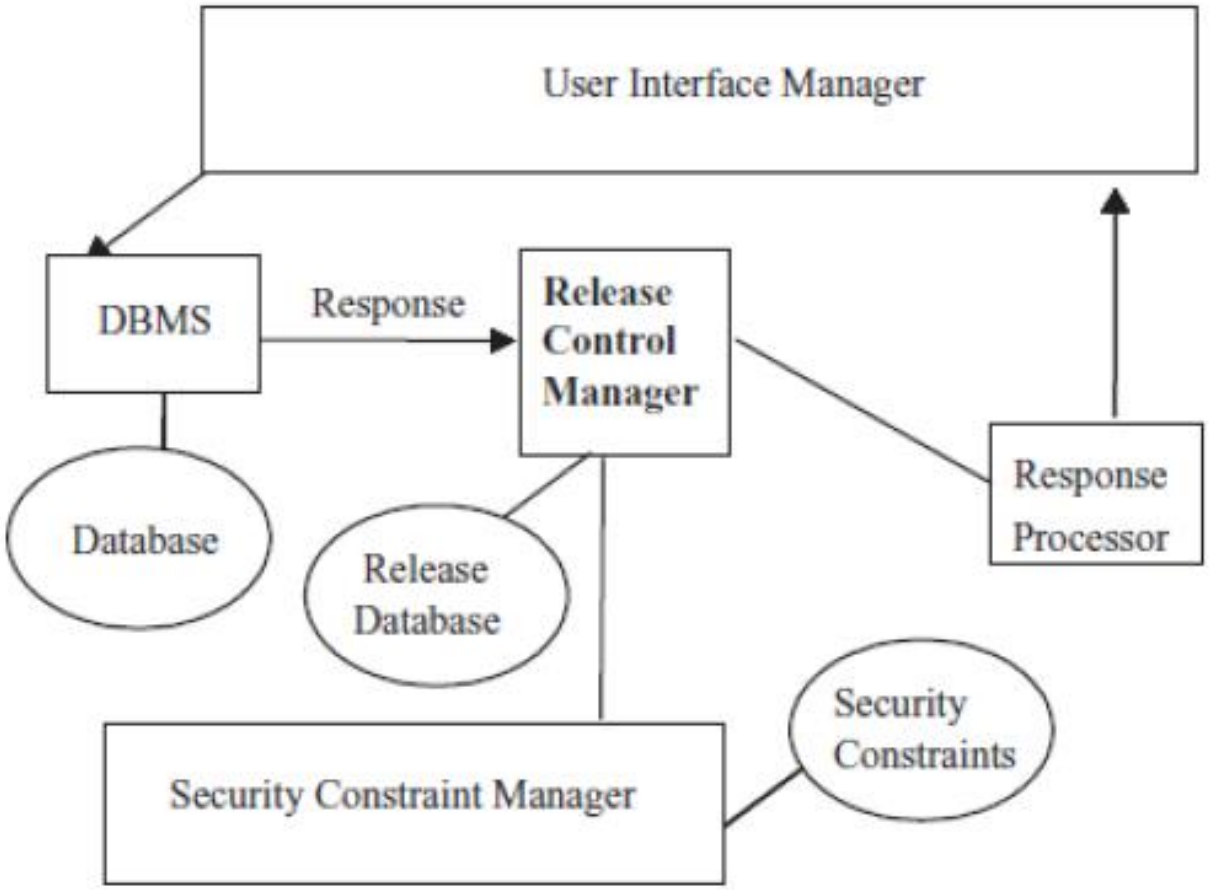

Figure 1. *Release Control Manager*

3. Conceptual structures for inference control (1)

- Before **designing the database,** we must examine the application and determine potential security breaches.
- **Semantic data models** are required, such as:
  - **semantic networks**
  - **conceptual graphs**
- They allow us to **model** an application and perform **reasoning** on it.
- We will analyze the use of conceptual structures for **inference control**.

3. Conceptual structures for inference control (2)

- Certain types of semantic data models, such as **Sowa's** (1984) **conceptual graphs**, have proven to be as powerful as the first-order logic.
- It has also been shown that conceptual graphs can be naturally extended to address mode and time problems.

3. Conceptual structures for inference control (3)

- **The main motivation** for using semantic data models is the following:

1. The use of semantic data models to represent applications is **consistent with the way how people perceive the world**.
2. It is more convenient, when the application is **analyzed manually**, for it to be represented as a graph (instead of a table or in a certain language).
3. A representation of the application using semantic data models can be used as a **front-end subsystem** of a system based on logic programming.
4. Reasoning strategies for representations based on semantic data models are **developed**.

3. Conceptual structures for inference control (4)

- The use of semantic data models to **address the inference problem** was first proposed by Hinke (1988).
- He proposed **the use of graphs to represent the application** and showed how inference can be detected by crossing alternate paths between two nodes in the graph.
- This technique allows the **detection of simple inferences through the transitivity property**, but it is not possible to detect the more complex ones.

3.1 Semantic networks and the inference problem (1)

- Semantic data models are interesting because they can be used to represent a **multilevel application**.
- Such a representation may be used by the SSO to **manually analyze the application**, in order to ensure that users cannot make unauthorized inferences.
- On the other hand, we can build a system that processes knowledge using strategies that have been developed for semantic data models and that automatically perform **analysis on application security.**

3.1 Semantic networks and the inference problem (2)

- Semantic networks are **simple models** having the power **to represent reasoning**.
- Standard semantic networks cannot represent multilevel applications, so **extensions** have been introduced.
- We consider **a semantic network to be a collection of nodes connected by edges**. Nodes represent **concepts, entities** etc., and edges represent **relations** between them.

3.2 Multilevel Semantic Networks (1)

- A Multilevel Semantic Network (MSN) is a **semantic network whose nodes and edges are classified at different levels of security**.
- Figure 2 presents some simple multilevel semantic networks.
- We assume that there are only two levels of security, **Unclassified** and **Secret**.
- This can be extended to include multiple levels of security.
- We assume that the bold lines and shapes are on the Secret level.

3.2 Multilevel Semantic Networks (2)

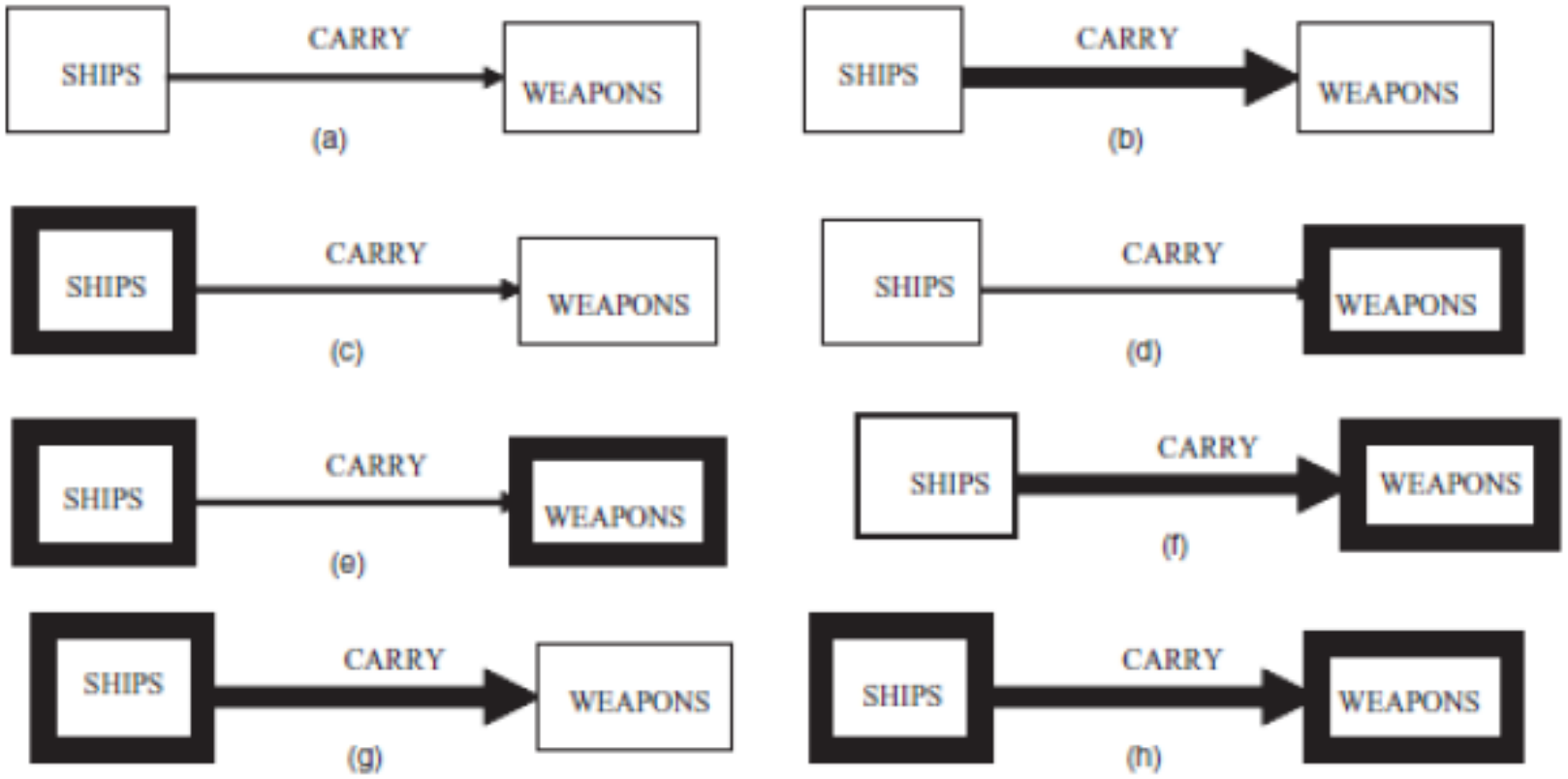

Figure 2. Multilevel semantic Networks

3.2 Multilevel Semantic Networks (3)

- Figure 2a) shows that ships carry weapons, and this information is Unclassified.
- In Figure 2b), ships and weapons are Unclassified, but the fact that ships carry weapons is Secret.
- In Figure 2c), Unclassified users know that an entity is transporting weapons, but they are not aware that ships are performing this transport.
- In 2d), users know that ships are carrying something, but they do not know that they are carrying weapons.
- In 2e) users know that a transport is done, but they do not know the carrier and the objects.
- In 2f), users have information about ships but know nothing about weapons.
- The situation is reversed in figure 2g), and in figure 2h) nothing is visible to Unclassified users.

3.2 Multilevel Semantic Networks (4)

- It is necessary to determine whether all the edges described in figure 2 are allowed.
  - For example, it could make sense to classify an edge at a level that dominates the levels of the nodes associated with the edge (the level of the "carry" relationship must dominate the levels corresponding to weapons and ships).
- Figure 3 shows a more elaborate multilevel semantic network. The interpretation from the Unclassified perspective of this Figure is as follows: Reagan carries passengers. Its captain is Smith and he has 20 years of experience. The ship was located in the Mediterranean Sea on June 16, 2000, and its destination was Italy.

3.2 Multilevel Semantic Networks (5)

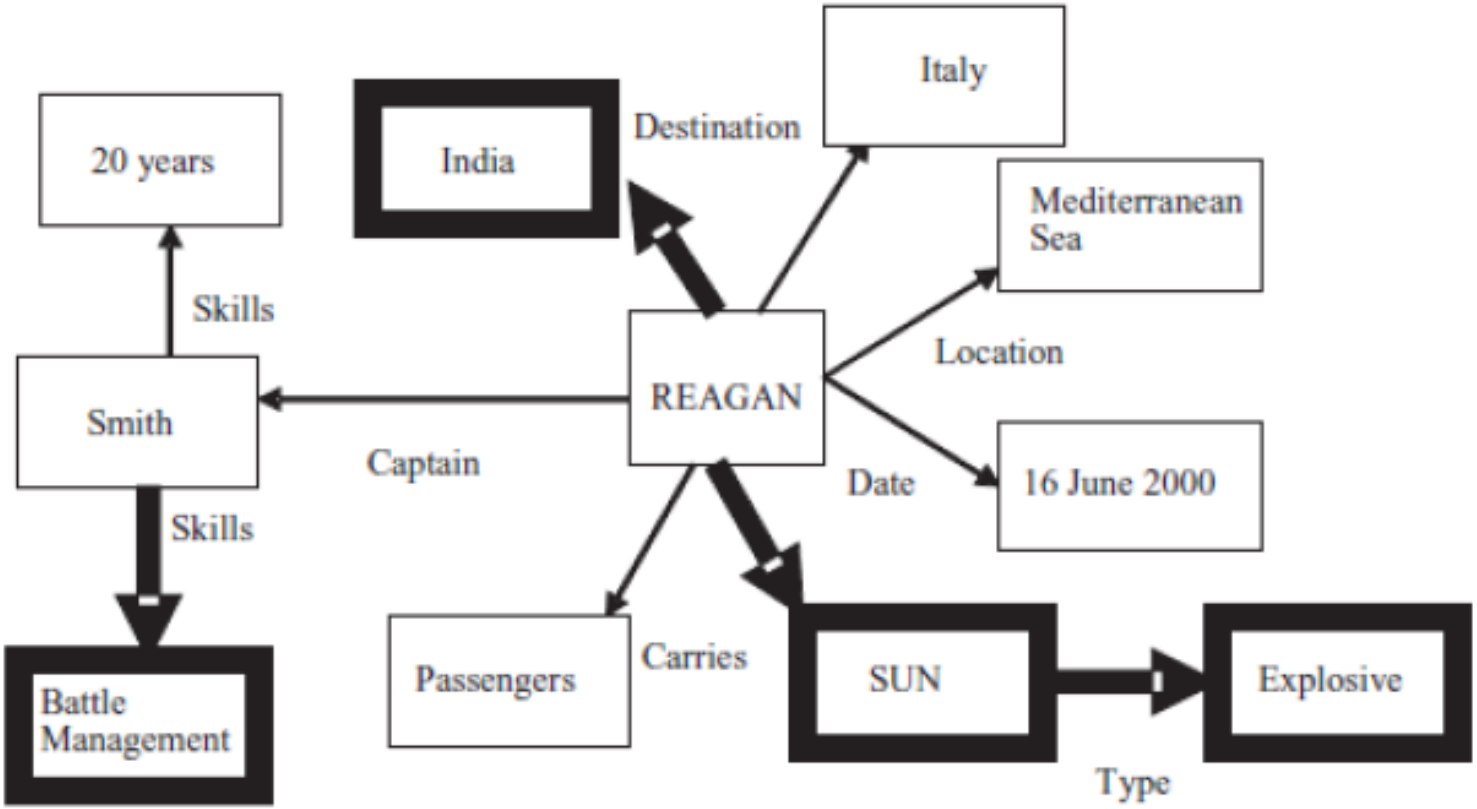

Figure 3. A complex Multilevel Semantic Network

3.2 Multilevel Semantic Networks (6)

- The interpretation from the Secret perspective is as follows: Reagan carries SUN, which is an explosive. Its captain is Smith who has battle experience. The ship was located in the Mediterranean Sea on June 16, 2000, and its destination was India.

- It can be seen that some **information** is **polyinstantiated**.
- Polyinstantiation occurs when users at different security levels **view differently** the same concept, entity, event, or relationship from the real-world.
- This is the mechanism used to "**hide**" information.

3.2 Multilevel Semantic Networks (7)

- The edges defined in the semantic networks considered so far illustrate specific relationships. In addition to these edges, a semantic network has **two types of standard edges: ISA and AKO**.
- An **ISA** edge specifies that a particular individual **belongs** to a specific group. Figure 4a) shows an ISA edge in which Reagan is defined as a particular type of ship, such as a battleship.
- An **AKO** edge defines a **subset** of a collection. Figure 4b) defines the ship collection as a subset of the water vehicle collection.

Figure 4. ISA, AKO edges

3.2 Multilevel Semantic Networks (8)

- It doesn't make sense to classify Reagan as Unclassified and Ship as Secret, because Reagan is an instantiation of Ship.
- By classifying Ship at the Secret level, we implicitly assume that any ship must be classified at least at the Secret level.
- It would also not make sense to classify Ship as Unclassified and Water Vehicle as Secret, because classifying Ship as Unclassified would imply that Water Vehicle should be classified at most as Unclassified.
- The classification of the Ship at the Secret level assumes implicitly that any Water Vehicle should be classified at most at the Secret level.

3.2 Multilevel Semantic Networks (9)

- Therefore, the following rules are required to ensure consistency:

- **A1.** If X ISA Y, then Level(X) >= Level(Y)
- **A2.** If X AKO Y, then Level(X) >= Level(Y)

3.3 Reasoning with Multilevel Semantic Networks (1)

- To achieve **reasoning** we need **rules**, which we will describe below.
- Most real-world applications work with large amounts of information, and a semantic network in which all information is captured would be extremely complex.
- We need a **minimal semantic network** with a **set of reasoning strategies** so that other information, called implicit information, can be deduced.
- For an application, the level of implicit information that can be deduced by a user at the *L* level should be dominated by *L*.

3.3 Reasoning with Multilevel Semantic Networks (2)

- Some rules for deducing implicit information are:

A3. If X AKO Y and Y AKO Z then X AKO Z. The level of the AKO edge from X to Z is the maximum of the levels of the AKO edges between X, Y and Y, Z

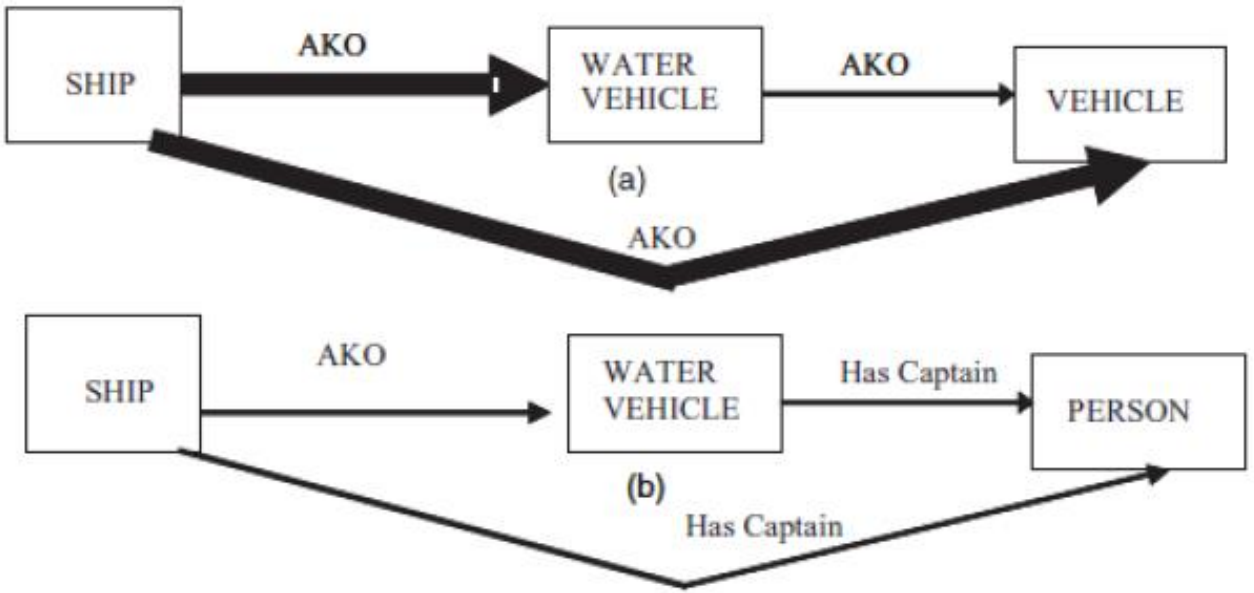

Figure 5 a), b)

3.3 Reasoning with Multilevel Semantic Networks (3)

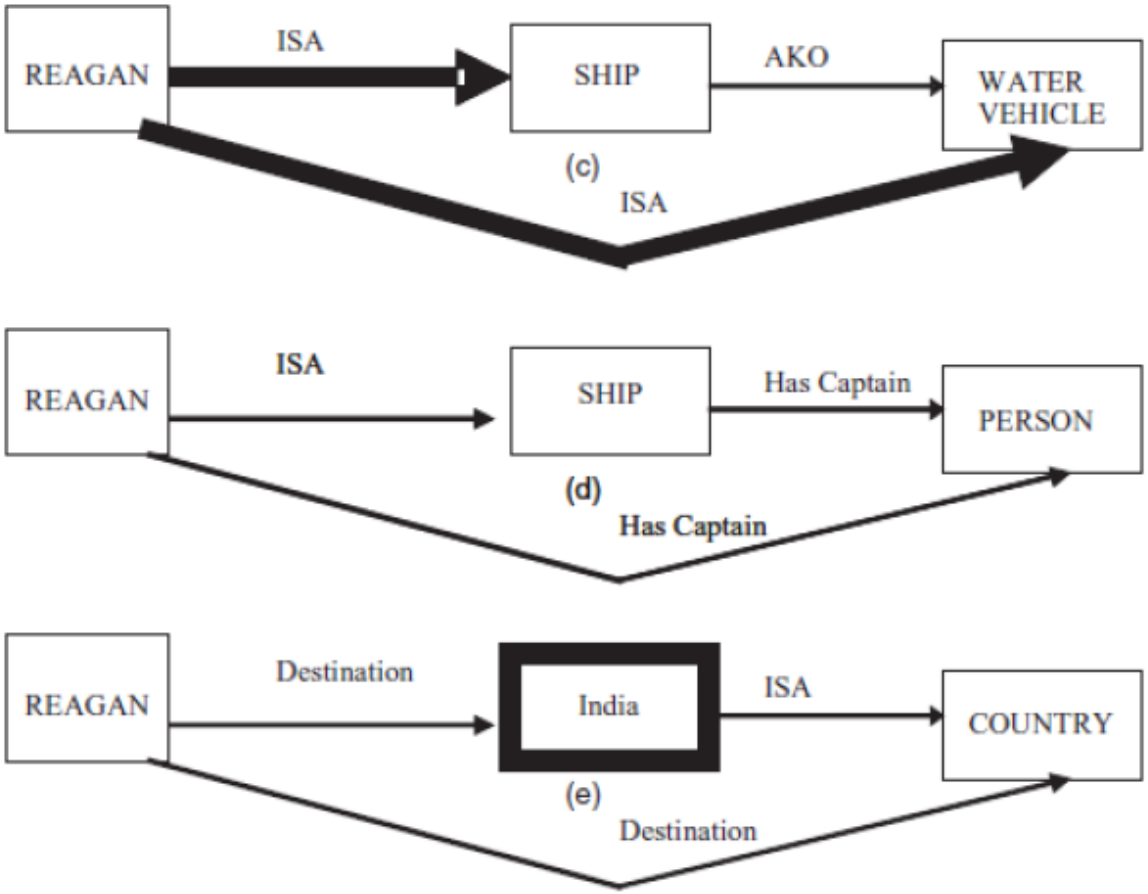

Figure 5c) – e). Examples of rules

3.3 Reasoning with Multilevel Semantic Networks (4)

- In figure 5a), the semantic network has the edges Ship AKO Water Vehicle and Water Vehicle AKO Vehicle.
- The AKO edge between the Ship and the Water Vehicle is Secret.
- Then, at the Secret level, it can be concluded that Ship AKO Vehicle.

3.3 Reasoning with Multilevel Semantic Networks (5)

- **A4.** If X AKO Y and Y is in the relation R with Z, then X is in the relation R with Z. The level of the relation R between X and Z is equal to the maximum value of the levels of the edge AKO between X, Y and the relation R between Y, Z.
- In Figure 5b), the semantic network contains “Ship AKO Water Vehicle” and “Water Vehicle has captain Person”. Then “Ship has captain Person”.
- **A5.** If X ISA Y and Y AKO Z, then X ISA Z. The level of the ISA edge between X, Z is equal to the maximum value of the levels of the AKO edge between Y, Z and the ISA edge between X, Y.
- In Figure 5c), Reagan ISA Ship. This edge is Secret. Ship AKO Water Vehicle, therefore there is an ISA edge at the Secret level between Reagan and Water Vehicle.

3.3 Reasoning with Multilevel Semantic Networks (6)

- **A6.** If X ISA Y and Y is in the relation R with Z, then X is in the relation R with Z. The level of the relation R between X and Z is equal to the maximum value of the levels of the edge AKO between X, Y and of the relation R between Y, Z.
- In figure 5d), the semantic network has the relation Reagan ISA Ship. Ship has captain Person. Therefore, Reagan has captain Person.

- **A7.** If X ISA Y and Z is in the relation R with X, then Z is in the relation R with Y. The level of the relation R that exists between Z and Y is equal to the maximum value of the levels of the ISA relation between X, Y and the relation R between Z, X.

3.4 Conditional statements and auxiliary networks (1)

- **Conditional statements** are in the form:

A if B1 and B2 and B3 and ... and Bn, where B1, B2, ..., Bn are the antecedents and A is the consequence.

- Conditional statements represent **clauses in a logical program**.
- A conditional statement can be **represented by auxiliary semantic networks**.
- Consider the following conditional statement:

Reagan's destination is India if the ship is located in the Mediterranean Sea and carries SUN, which is an explosive material.

3.4 Conditional statements and auxiliary networks (2)

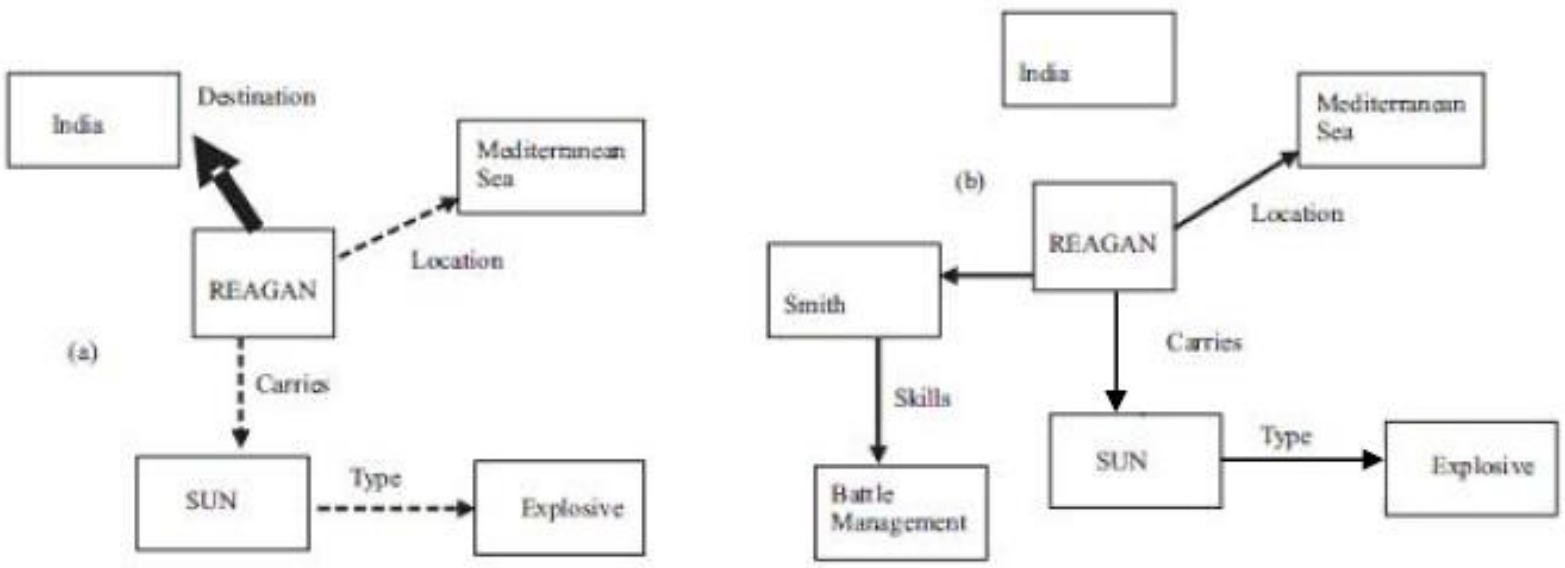

Figure 6a), b)

3.4 Conditional statements and auxiliary networks (3)

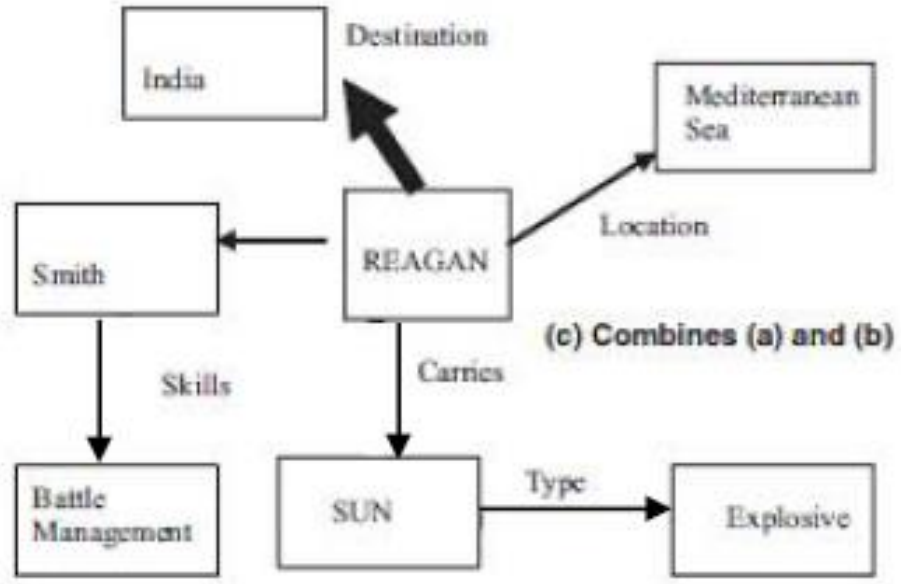

Figure 6c) The application of the transfer rule

3.4 Conditional statements and auxiliary networks (4)

- The transfer rule is applied to process conditional statements:

A8 (**The Transfer Rule**). If **all the dotted lines in the auxiliary network** are presented as **solid lines in a multilevel semantic main network**, and the **level** of each solid line in the main network dominates the level of the corresponding dotted line in the auxiliary network, then the solid line in the auxiliary network is drawn as a solid line in the main network. The security level of the drawn line is the maximum value of the levels of all the lines in the auxiliary network and all corresponding continuous lines, which are already in the main network.

3.4 Conditional statements and auxiliary networks (5)

- The conditional statement is represented by the auxiliary network shown in figure 6a). The **conditions** are represented by **dotted lines**, and the **conclusion** by **solid lines**.
- Figure 6b) shows that the dotted lines in the auxiliary network appear as solid lines in the multilevel semantic network.
- Figure 6c) shows that the solid line in the auxiliary network is added to the multilevel semantic network at the appropriate security level.

4. Enforcing the security constraints (1)

- As we stated in the previous course, security constraints are rules that assign security levels to data. We will represent security constraints through "**constraint networks**".

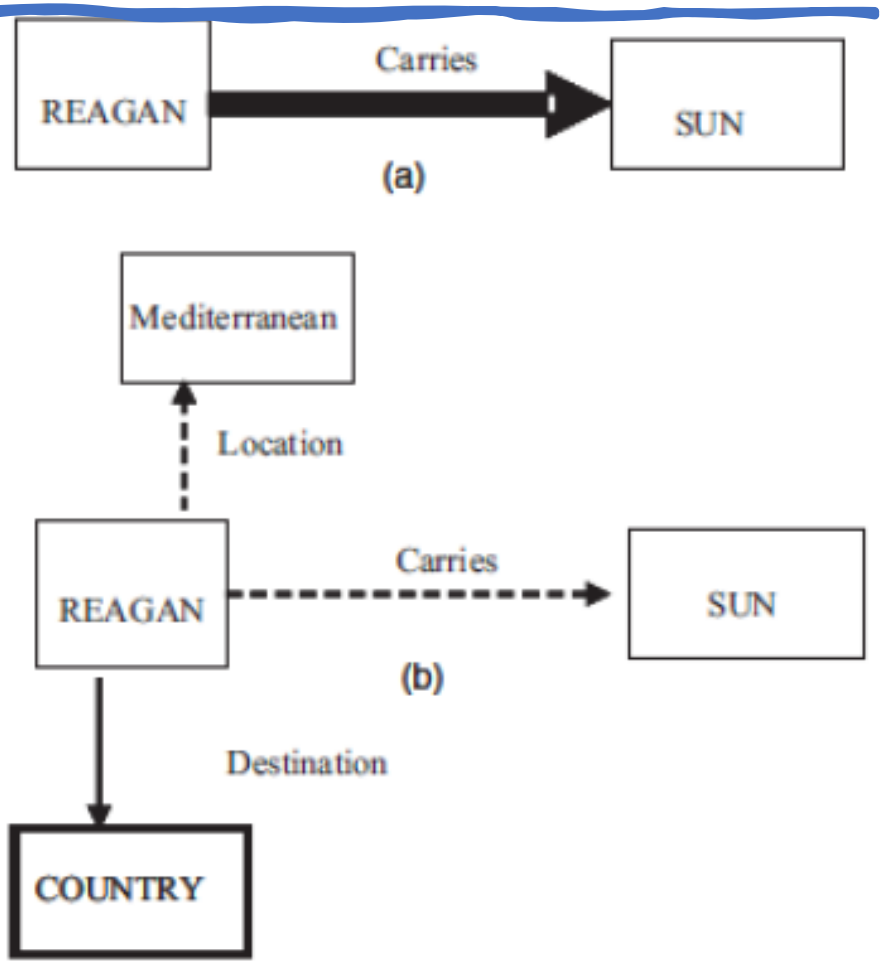

Figure 7. Representing the security constraints

4. Enforcing the security constraints (2)

- A **constraint network** is a semantic network, main or auxiliary, that specifies only constraints.
- Although semantic networks are generally used to represent application-specific information, semantic networks for constraints are used to **represent security constraints** so that any security breaches in the application can be detected.
- Similarly, auxiliary semantic networks are used to deduce implicit information, and security constraints represented as auxiliary semantic networks are used to **detect security breaches**.
- Therefore, a distinction is made between simple auxiliary networks and those related to constraints.

4. Enforcing the security constraints (3)

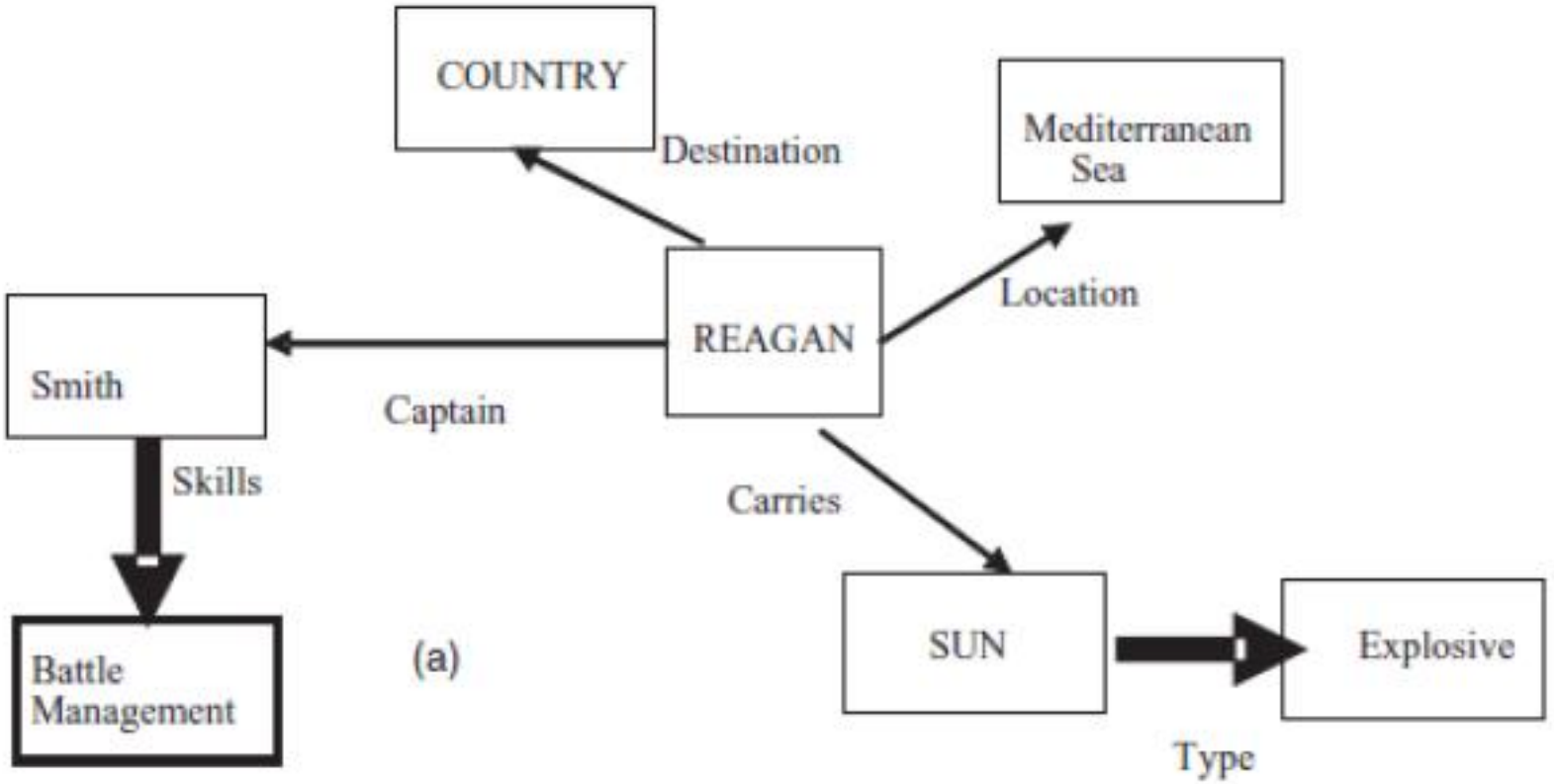

Figure 8a)

4. Enforcing the security constraints (4)

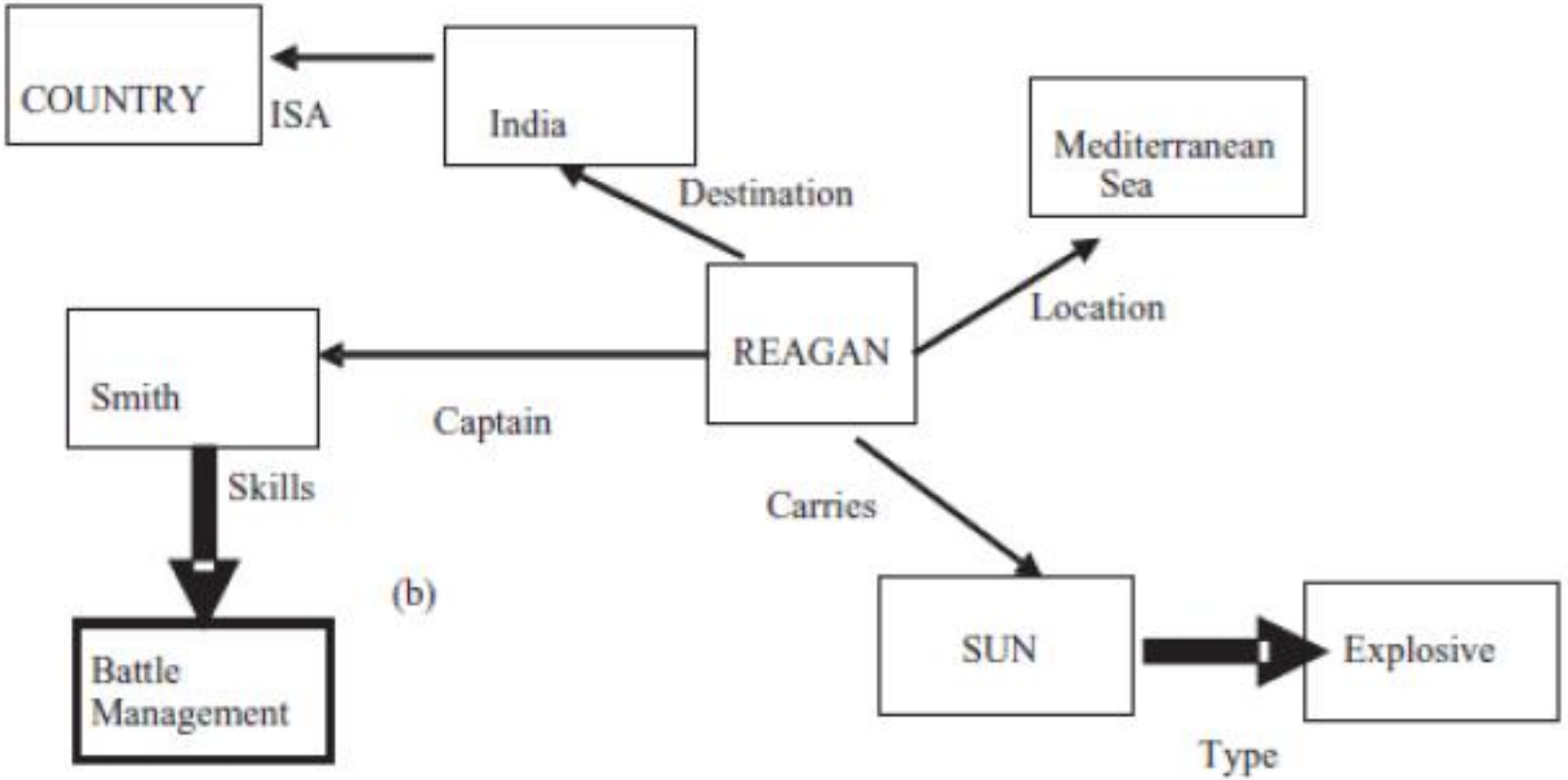

Figure 8b) Failure to comply with security constraints

4. Enforcing the security constraints (5)

- Figure 7a) classifies the fact that Reagan performs transports at the Secret level.
- Figure 7b) shows a constraint that classifies Reagan's destination country at the Secret level, if Reagan is located in the Mediterranean Sea and carries SUN.
- Security breaches occur (directly or indirectly) **if the network of constraints contradicts the multilevel semantic network that represents the application** (directly or indirectly).
- The semantic network in Figure 8a) directly violates both constraints in Figure 7.
- In Figure 8a), the fact that Reagan carries something is not Secret, which directly violates the constraint in Figure 7a).
- Also, in Figure 8a), Reagan is located in the Mediterranean Sea and carries SUN. Its destination country is Unclassified, which directly violates the constraint in Figure 7b).

4. Enforcing the security constraints (3)

- Non-compliance with the constraints may also occur indirectly.
- This situation occurs when the **implicit information contradicts the security constraints**.
- Figure 8b) shows how the security constraint in Figure 7b) is violated indirectly. Here, Reagan carries SUN and is located in the Mediterranean Sea. Its destination country, India, is Unclassified. On the other hand, from rule A7, it would follow that Reagan's destination should be Secret. Therefore, the constraint in Figure 7b) is indirectly violated.