engineering proceedings

MDPI

*Proceeding Paper*

# Usage of OLAP Cubes as a Data Model for DSS †

**Nikolai Scerbakov [1], Alexander Schukin [2] and Eugenia Rezedinova [2,*]**

[1] Institute of Interactive Systems and Data Science, Graz University of Technology, 8010 Graz, Austria; scerbakov@tugraz.at
[2] Institute of Computer Science and Cybersecurity, Peter the Great St. Petersburg Polytechnic University, St. Petersburg 195251, Russia; alexander.schukin@spbstu.ru
* Correspondence: rezedinova_eyu@spbstu.ru
† Presented at the International Conference on Electronics, Engineering Physics and Earth Science (EEPES 2025), Alexandroupolis, Greece, 18–20 June 2025.

**Abstract**

A decision support system (DSS) is a software application designed to determine suitable actions for specific organizational situations. Its main component is a data repository analyzed to produce decisions. This paper describes the data organization (Data Model) as a multi-dimensional OLAP cube with amendments for decision-making support. We present DSS functionality as building (slicing) hyper-cubes into decision sub-cubes. The system's adjustment and evolution involve changing the granularity of these sub-cubes. We discuss the merging and splitting of hyper-cubes, arguing that this functionality is adequate for creating complex, real-time DSSs for various incidents, such as cybersecurity incidents.

**Keywords:** decision support systems; data model; OLAP cube

check for updates

Academic Editor: Grigor Mihaylov

Published: 22 August 2025

**Citation:** Scerbakov, N.; Schukin, A.; Rezedinova, E. Usage of OLAP Cubes as a Data Model for DSS. *Eng. Proc.* **2025**, *104*, 4. https://doi.org/10.3390/engproc2025104004

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Decision support systems (DSSs) indeed play a crucial role in modern management, especially in high-stakes, time-sensitive environments, where they prevent cybersecurity incidents. In this paper, we focus on the DSS that automatically provides a ready-to-run course of actions (scenario) tailored to a specific situation. This type of DSS is particularly important in scenarios where rapid decision-making is critical. Also, such systems often help to tackle different accidents, such as firefighting accidents, traffic accidents, ship accidents, and similar things [1–3].

Normally, the personnel responsible for tackling accidents use predefined scenarios and corresponding action plans. If we consider cybersecurity as an example and a threat is detected in the system, the predefined scenario may suggest deploying protective measures and isolating affected systems. When discussing predefined scenarios, the functionality related to the optimization, adaptation, and evaluation of different possible decisions can be replaced with the functionality of selecting the most suitable scenario from a reasonably large number of available options [4,5]. It is possible that the relevant scenario is simply missing, and there must be a well-defined mechanism for developing scenarios using a group of experts. The DSS oriented towards the use of predefined scenarios for decision-making in cybersecurity consists of the following components (see Figure 1).

Repository (database) of predefined scenarios: This database is intended to serve rather specific goal: it must search for the most relevant scenario for a concrete management case.

Use-case definition component: The system should automatically provide the most relevant scenario for a particular use case. Thus, the term "use case" must have a precise definition that might be used by the managerial team.

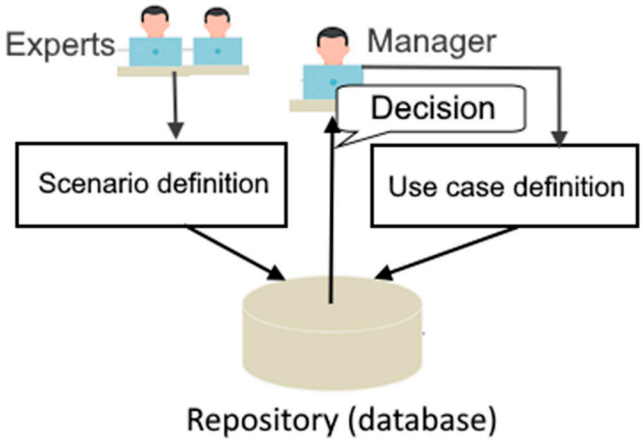



**Figure 1.** DSS architecture.

Scenario definition component: As the manager defines a use case, there may be situations where the system cannot recommend an existing predefined scenario. This case potentially may have two solutions: either the manager manually elaborates the course of actions, the resultant scenario is practically used, and the scenario is reviewed, refined, and added to the repository of predefined scenarios after the incident; or the DSS developers provide a clear mechanism for identifying use cases where the predefined scenarios do not exist. In the second case, there must be a mechanism for the preliminary development of missing scenarios.

## 2. Data Model

To be compatible, all the above-mentioned components of DSS must be based on a unified Data Model. The following features are necessary for the Data Model used in the implementation of the DSS [6–8]:

- It must be suitable for defining individual scenarios and storing information on a large number of predefined scenarios.
- It must be suitable for defining use cases.
- It must provide a mechanism for selecting scenarios suitable for a particular use case.
- It must provide a mechanism for identifying missing scenarios.

To design a Data Model for storing predefined scenarios in a cybersecurity application, we need to carefully consider the data structures and operations that will best support the tasks at hand. Here is a breakdown of how we can approach this:

The database is organized as a multi-dimensional space (cube), where each dimension represents an attribute defining a scenario [9,10]. For a cybersecurity application, the dimensions could include the following:

- Type of Threat (e.g., malware, phishing, data breach, DDoS attack);
- Vulnerability Level (e.g., low, medium, high);
- User Type (e.g., student, instructor, administrator);
- Device Type (e.g., PC, mobile device, platform);
- Attack Method (e.g., social engineering, exploitation of vulnerabilities, unauthorized access);
- Incident Time (e.g., morning, day, evening, night);
- Incident Type (e.g., hacking attempt, data leak, phishing);
- Security Resources (e.g., antivirus software, intrusion detection systems, firewalls).

Each dimension in the multi-dimensional space can be divided into discrete or continuous attribute scopes. These scopes represent meaningful ranges or categories for the

attribute. For example, the sub-cube for a DDoS attack on an administrator is represented as follows (Figure 2):

- Type of Threat: DDoS Attack;
- Vulnerability Level: Medium;
- User Type: Administrator;
- Device Type: Platform;
- Attack Method: Exploitation of Vulnerabilities;
- Incident Time: Day;
- Incident Type: Hacking Attempt;
- Security Resources: Intrusion Detection System.

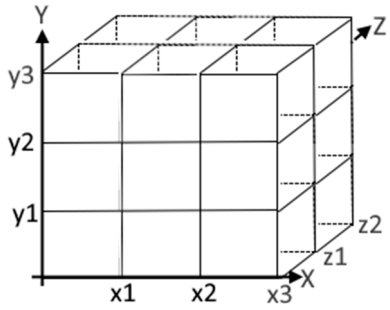



**Figure 2.** Sub-cubes in the multi-dimensional space.

A sub-cube is a smaller, well-defined region within the multi-dimensional space. It is formed by the combination of attribute scopes from each dimension. For example, a smaller sub-cube for a DDoS attack is represented as follows:

- Type of Threat: DDoS Attack;
- Vulnerability Level: High;
- User Type: Administrator;
- Device Type: PC;
- Attack Method: Exploitation of Vulnerabilities;
- Incident Time: Day;
- Incident Type: Hacking Attempt;
- Security Resources: Firewalls.

The model supports the following set of operations on the multi-dimensional space: The embracing sub-cube can be found by the coordinates of the object in the space. For example, the decision cube in Figure 3 is the embracing cube for the object with coordinates x, y, and z if x > x1, x <= x2, y > y1, y <= y2, z > z1, and z <= z2.

Thus, the decision associated with a decision cube can be extracted by the coordinates of any object inside the cube.

Sub-cubes having no associated decisions can be also extracted from the space.

The sub-cube represents a special room for the unique scenario (decision) applicable for the combination of attribute scopes. Therefore, we will call such sub-cubes "Decision Cubes" (see Figure 3).

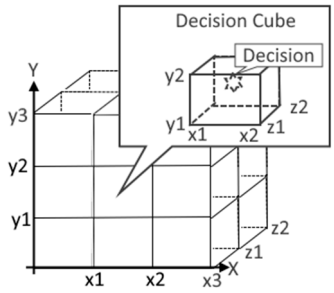



**Figure 3.** Decision cube.

Slicing is an operation that extracts a subset of decision cubes by providing a subset of object coordinates. For example, two sub-cubes with an identical z scope (see Figure 4) can be selected by providing two coordinates, x and y. Similarly, just an x coordinate identifies six sub-cubes with an identical x scope.

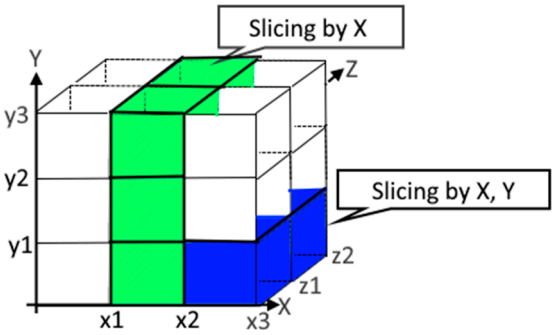



**Figure 4.** Slicing the cube.

## 3. System Functionality

The recent archive of known managerial cases is analyzed and digitalized to build a first version of the multi-dimensional space. Real-world accidents are represented digitally using a predefined set of parameters (coordinates). These coordinates map each case to a specific location within the multi-dimensional decision cube.

Optionally, system administrators may analyze the distribution of ready-to-run scenarios within the cube related to the security of the LMS (Learning Management System). If there is a significant number of decision sub-cubes without associated scenarios, the administration may invite experts to develop scenarios for such hypothetical cases. Note that the coordinates of each decision sub-cube provide a reasonable description of the hypothetical case. For example, the input information for the experts may sound like this: Develop a course of action for responding to a DDoS attack on an LMS used for online learning, taking into account attack protection and ensuring the availability of educational materials for students.

Once the preliminary scenario is ready, it is stored in the database to cover the missing decisions. When a real security threat needs to be addressed, the case is also digitized using the same set of parameters as the decisions. Thus, each case receives coordinates x, y, z, etc. The coordinates refer to a decision cube. There can be two situations: either the cube has an associated decision, or it does not. If a decision is found, the manager is provided with all the relevant documents. If the decision sub-cube does not have an associated decision, the search continues until a suitable decision is identified or the search space is exhausted.

The system slices the cube by all the coordinates and checks the next sub-cubes in each direction. If a decision is found, it is offered to the manager; if not, the process is repeated

for two neighboring sub-cubes, and so on. The process is based on the implicit assumption that solutions for similar scenarios can be adapted to the current situation. For example, a good solution for an LMS serving 1000 students may be suitable for a system serving 2000 students with minor adjustments. A solution designed for protecting one platform may also work for another platform with some modifications.

## 4. Post-Accident Adjustment

The assumption that similar cases (i.e., cases that fall into one and the same decision sub-cube) can be treated by a single scenario may not always hold true. The effectiveness of the process depends on the granularity, quality, and completeness of the whole multi-dimensional space. To adjust the space by accommodating more and more use cases, the following process is applied: After each application of the DSS, the system administration invites several experts to evaluate the results of the application. Potentially, there may be three situations:

- The decision is fully applicable to the case.
- The decision is partially applicable after making some modifications.
- The decision is not applicable to the case, and development of the new scenario is needed.

If the recommended decision is found to be fully applicable, there are no additional adjustments needed, and the decision receives additional positive points as a grade.

If the decision is only partially applicable, the scenario is amended with the comments reflecting possible modifications of the scenario for serving special cases.

If the system produces an entirely wrong decision, the system administration should especially analyze the case. If the scenario that was associated with the case (decision sub-cube) was not previously successful, it may be simply replaced with the new scenario that was developed for this case. If the scenario was previously used successfully and received a positive grading, the cube granularity must be enhanced. In this case, the coordinates of the failed case are compared with the coordinates of the case that was solved successfully, and one or more coordinate ranges are divided into smaller, more precise scopes. This process creates new smaller decision sub-cubes where old and new scenarios can be separated and associated in different decision sub-cubes.

Similarly, the scopes can be merged if the decision was found in one of the near sub-cubes by means of the slicing procedure, and the decision was assessed as fully suitable for the case by the administration.

## 5. Post-Accident Adjustment

The model is structured using three MongoDB collections: Resources, UseCases, and Cube [11]. UseCases collection: This collection stores well-structured JSON documents adhering to a predefined schema. Each document includes attributes such as the following:

- _id (ObjectId);
- Type of Threat (string);
- Vulnerability Level (integer);
- Attack Method (ObjectId, referencing Resources);
- Incident Time (Date);
- Incident Type (ObjectId, referencing Resources).

Resources collection: This collection contains supplementary data (e.g., attack methods, incident types, countermeasures) in flexible JSON formats.

Cube collection: This collection represents the OLAP cube as a set of JSON documents, each defining a sub-cube. Sub-cubes are characterized by dimension ranges, where each

original attribute is replaced with Min and Max pairs (e.g., Min Vulnerability Level and Max Vulnerability Level). For non-numerical attributes (e.g., Incident Type), these values may be identical.

The model is supposed to support the following Data Operations that are implemented via the Aggregation Pipelines:

Sub-Cube Retrieval

Locates a sub-cube containing a specific use case by matching dimension bounds.

```
db.Cube.find({
$and: [
{ "Max Vulnerability Level": { $lte: value } },
{ "Min Vulnerability Level": { $gte: value } },
// . . . additional dimension filters
]
});
```

Drill-Up (Summarization)

Aggregates data (e.g., counts scenarios for a specific incident type).

```
db.UseCases.aggregate([
{ $match: { "Incident Type": "SQL Injection" } },
{ $group: { _id: "$Incident Type", totalScenarios: { $sum: 1 } } }
]);
```

Drill-Down (Granular Analysis)

Breaks down data by finer criteria (e.g., threats by attack type and vulnerability level).

```
db.UseCases.aggregate([
{ $match: { "Attack Type": "DDOS", "Vulnerability Level": { $gt: 2 } } },
{ $group: { _id: "$Incident Type", count: { $sum: 1 } } }
]);
```

Slice (Single-Layer Extraction)

Extracts data for a specific dimension value (e.g., all use cases for an attack type).

```
db.UseCases.aggregate([
{ $match: { "Attack Type": "DDOS" } }
]);
```

## 6. Conclusions

In this paper, we described the application of a Data Model built around the OLAP cube (multi-dimensional space) for decision support systems. We introduced the concept of a decision sub-cube for structuring a large repository of ready-to-run scenarios that can be used for making decisions in different situations (use cases).

We also introduced a special algorithm for recursive searching decisions around a certain use case that is based on the operations of slicing the cube. The DSS outcome is greatly depends on the quality of archived scenarios (prepared decisions) and granularity (specificity of the decisions). The model supports a special mechanism for adjusting granularity by continuously evaluating outcomes. By refining scenarios and enhancing granularity, the system adapts to new challenges and improves its decision-making capabilities over time. In this sense, we can say that the system learns from the results of practical application. However, administration should support a balance between the complexity and practicality of the system manually.

This approach provides a robust framework for the practical implementation of a DSS in different areas of management. We describe one possible practical implementation that is based on the mapping the OLAP cube onto Mongo DB collections, and mapping such OLAP cube-specific operations onto the Mongo DB aggregation pipelines.

While OLAP for DSSs is well-studied, our innovations lie in the following:

- Sub-cube structuring for scenario management.
- Self-optimizing granularity through feedback.
- Reproducible NoSQL design, facilitating adoption.

The system's effectiveness depends on scenario quality and granularity, requiring manual oversight to balance complexity and usability. This framework is adaptable across diverse management domains.

**Author Contributions:** Conceptualization, N.S. and A.S.; methodology, E.R.; software, N.S.; validation, N.S., A.S. and E.R.; formal analysis, A.S.; investigation, E.R.; resources, N.S.; data curation, A.S.; writing—original draft preparation, N.S.; writing—review and editing, E.R.; visualization, A.S.; supervision, N.S.; project administration, E.R.; funding acquisition, A.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Dataset available on request from the authors.

**Acknowledgments:** During the preparation of this manuscript, the authors used GPT-4o mini for the purposes of improving the English language, as the authors are not native speakers. This tool assisted in enhancing the clarity, coherence, and overall quality of the text, ensuring that the manuscript met the standards expected in academic writing. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Ergun, H.; Ghlonti, G.; Rodonaia, I. Advancing University Management Decision-Making Through Data Warehousing and OLAP-Driven Decision Support Systems. *Int. J. Sci. Res.* **2024**, *13*, 850–856. [CrossRef]
2. French, S.; Turoff, M. Decision support systems. *Commun. ACM* **2007**, *50*, 39–40. [CrossRef]
3. Mannina, G.; Rebouças, T.F.; Cosenza, A.; Sànchez-Marrè, M.; Gibert, K. Decision support systems (DSS) for wastewater treatment plants—A review of the state of the art. *Bioresour. Technol.* **2019**, *290*, 121814. [CrossRef] [PubMed]
4. Ankana; Dhanaraj, G. Study of selected influential criteria on groundwater potential storage using geospatial technology and multi-criteria decision analysis (MCDA) approach: A case study. *Egypt. J. Remote Sens. Space Sci.* **2021**, *24*, 649–658. [CrossRef]
5. Cunha, D.A.; Andrade, M.; Prado, L.A.; Santana, L.O.; Gonçalves da Silva, M.P. RISK assessment in airport maintenance runway condition using MCDA-C. *J. Air Transp. Manag.* **2021**, *90*, 101948. [CrossRef]
6. Voznytsia, A.; Kashkevych, S.; Radchenko, K.; Krant, D.; Buyalo, O.; Matsyi, O. Decision Support Systems: Mathematical Support. In *Collective Monograph*; Kashkevych, S., Ed.; TECHNOLOGY CENTER PC.: Kharkiv, Ukraine, 2025. [CrossRef]
7. Ravat, F.; Teste, O.; Tournier, R.; Zurfluh, G. Finding an application-appropriate model for XML data warehouses. *Inf. Syst.* **2010**, *35*, 662–687. [CrossRef]
8. Cyganiak, R.; Reynolds, D. The RDF Data Cube Vocabulary. W3C Recommendation. 16 January 2014. Available online: https://www.w3.org/TR/vocab-data-cube/ (accessed on 8 April 2025).
9. Reynolds, D. The Organization Ontology. W3C Candidate Recommendation. 25 June 2013. Available online: http://www.w3.org/TR/vocab-org/ (accessed on 8 April 2025).
10. Yu, R.; Fu, Y. Advancing Image Database Technologies in Clinical Decision Support Systems: A Systematic Literature Review. *Int. J. Eng. Sci. Technol.* **2024**, *6*, 132–145. [CrossRef]
11. Djiroun, R.; Boukhalfa, K.; Alimazighi, Z. Designing data cubes in OLAP systems: A decision makers' requirements-based approach. *Cluster Comput.* **2019**, *22*, 783–803. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.