A community-driven database for the dynamic representation of approaches, processes, methods, and tools for multidisciplinary product development

Muhammad Bilal [1,2], **Benoît Eynard** [3] **and Julia Guérineau** [1,✉]

[1] École de technologie supérieure, Canada, [2] National University of Sciences and Technology, Pakistan,
[3] Université de technologie de Compiègne, France

✉ julia.guerineau@etsmtl.ca

ABSTRACT: The advent of multidisciplinary product development may require a corresponding evolution or adaptation of product development practices within companies. To support this, researchers have developed various groupings of concepts and techniques, such as "toolboxes" or "maps", which can be assimilated to static databases. Consequently, this article presents a first step towards a community-driven database for the dynamic representation of links between approaches, processes, methods and tools in research documents. Following a comparative analysis of different representations, a preliminary design of a dynamic database is presented using Unified Modeling Language models to define its architecture. A use case diagram paired with screenshots of the dynamic database presents the core functionalities, which include real-time data filtering, visualisation, navigation and modification.

KEYWORDS: concepts and techniques repository, graph database, information management, organisation of product development, visualisation

1. Introduction: organising concepts and techniques

Over the past two decades, there has been a notable increase in the integration of connectivity and digital technologies into products. The resulting development is organised around multidisciplinary teams combining mechanical, electrical, electronic and software engineering. For companies, this implies a significant technical and organisational complexity, which has an impact on all stakeholders of the extended enterprise (Zhang & Thomson, 2016). In this context, it may be necessary for companies to adapt or evolve their product development practices. This adaptation or evolution can be conducted at different levels of product development (Guérineau et al., 2018), while ensuring coherence across them. In order to support companies in their multidisciplinary product development (MPD), several researchers have conceptualised a "toolbox", a "collection", a "catalogue", or a "repository" of methods and tools, some of which are positioned along a product development process model (Böhmer, 2018; Goevert & Lindemann, 2018; Miranda et al., 2017). Amongst the most recent ones, Guérineau et al. (2022) proposed three "maps", each one respectively organising a variety of concepts and techniques from the scientific literature for mechatronics (Bradley & Hehenberger, 2016), cyber-physical systems (Baheti & Gill, 2011), and "smart" products (Kortuem et al., 2007) development. In the context of this article, the expression "concepts and techniques" groups the variety of approaches, processes, methods and tools for product development. The three maps are organised on the basis of a four-level model from the same authors (Guérineau et al., 2018), providing a hierarchical structure for organising the concepts and techniques. Each level, namely Approach, Process, Method and Tool, plays a specific role in guiding and supporting product development.

The maps from Guérineau et al. (2022) offer a comprehensive view on the multiple concepts and techniques discussed by the scientific literature for MPD and their associations. Through this visual representation, researchers and industry practitioners can navigate and understand the relationships

between them, especially their combinations and joint use. However, the authors acknowledged the static nature of the representation and its associated limitations. Primarily, these limitations involve the filtering of maps and the updating of information over time. To address these two limitations, the necessity for a "dynamic and navigable representation" and a "community-based and maintained database" has been emphasised (Guérineau et al., 2022). It is anticipated that these two directions will resonate with both the scientific community and the industry. Accordingly, the main objectives of this project are (1) to develop a dynamic graphical representation that will enable researchers and industry to navigate, visualise, and analyse the associated links established between the concepts and techniques identified in the research papers, and (2) to enable the community to contribute actively by adding and updating concepts and techniques, their links and the associated scientific references.

The article builds on previous research and addresses two of its main limitations, while reusing the underlying codes and semantics. The next section outlines the sub-objectives and criteria for comparing three potential dynamic representations: ontologies, graph databases, and relational databases, leading to the selection of the graph database. Section 4 presents the design of the proposed dynamic database using Unified Modeling Language (UML) models, along with its core functionalities. The discussion section covers the implications of the proposed database, and future work, leading to the conclusion.

2. Design criteria for evaluating dynamic representations

To ensure the project's alignment with the two main objectives outlined above, these are refined into five sub-objectives. The latter address specific facets of the project, formalising the need for a dynamic representation, capturing and representing the semantics of the initial maps, and providing filtering and querying capabilities with the possibility of further analysis and extraction of query results. In addition, the updating of the maps' data and the usefulness of the representation for both industry and researchers are specified. The relationship between the sub-objectives and the main objectives is presented on the left side of Table 1 with the corresponding main objective number in brackets. Then, each sub-objective is derived into a design criterion. Each design criterion and its description are given on the right-hand side of Table 1. The adoption of a structured modus operandi, whereby the main objectives are broken down into sub-objectives, which in turn are paired with design criteria, ensures that the selected representation will support each individual facet of the project. This selection is discussed in the next section.

Table 1. Criteria used for the comparison and analysis of representations

Sub-objectives associated to Design Criteria (Associated Main Objectives)	Description of Design Criteria	Design Criteria Name
Enable the dynamic visual representation of concepts and techniques, and associated links between papers. (1)	Evaluates how well the representation displays the maps interactively, allowing users to view and navigate the visualisation in real-time.	Dynamic visualisation (C1)
Provide filtering/querying capabilities to explore different concepts and techniques, as well as their associated links. (1)	Assesses the efficiency, speed, and accuracy of querying/filtering techniques to retrieve specific information from the representation.	Querying capability (C2)
Ensure an appropriate representation of semantic knowledge from papers, i.e., retainment of associated links and the four-level model. (1)	Assesses how effectively the representation captures and conveys the meaning of data, including associated links between concepts and techniques.	Semantic Representation (C3)
Provide researchers with an accessible means of analysing, extracting, and updating data. (1) & (2)	Evaluates the likelihood of researchers adopting the representation by reviewing its prevalence and relevance in the scientific literature, i.e., how often it is used in research.	Adoption by the research community (C4)
Ensure the representation is appropriate and useful to the industry. (1) & (2)	Assesses the practical benefits of the representation for industry based on qualitative and practical insights.	Perceived value to industry (C5)

3. Comparison and analysis of representations

Based on the design criteria defined in the previous section, a comparison and analysis of the potential dynamic representations are presented below. The next sections give an overview of each of them, i.e., ontologies, graph databases and relational databases, before discussing the one that best meets the defined design criteria. Representations were chosen based on their similarity to the intended static structure. Other representations like Resource Description Format (RDF) were considered but deemed unsuitable due to its deviation from simple node-link structures (Ravat et al., 2020). Each design criterion is evaluated on a qualitative basis, with equal weightage given to each.

3.1. Ontologies

Ontologies are widely adopted in research for knowledge representation and semantic modelling (C4) (Martinez-Cruz et al., 2012; Sir et al., 2015). Indeed, ontologies excel in capturing and conveying semantic information (C3) (Martinez-Cruz et al., 2012) effectively representing complex relationships between entities. They also facilitate sophisticated querying using semantic query languages like SPARQL, which allows for advanced searches and filtering based on semantic relations (C2). As noted by Figueres-Esteban et al. (2018), ontologies are more suited for static representations and may struggle with dynamic visualisation (C1). To address this limitation Chen & Xing (2022) suggest that visualisation tools may be necessary to create interactive views. Finally, their use in specific industry applications is primarily limited to scenarios that require structured knowledge representation (C5).

3.2. Graph databases

Graph databases perform well in dynamic visualisation (C1) by representing data as nodes and relationships, supporting interactive exploration and real-time navigation effectively (Ravat et al., 2020). They can also provide fast, efficient and flexible querying capabilities (C2). Their ability to efficiently traverse relationships makes them well suited to handle interconnected data (Chen & Xing, 2022). Graph databases face practical limitations, including limited declarative querying, which hampers optimisation and performance on large-scale data. Additionally, poor data partitioning can slow queries and create visualisation bottlenecks due to inadequate horizontal scalability (Pokorný, 2015). Although they do not offer the same formal semantics as ontologies, graph databases are still quite effective for visualising meaningful insights (C3). In addition, graph databases have an increasing utilisation in both academic and industry settings (C4, C5) (Figueres-Esteban et al., 2018). Providing scalable and flexible solutions for representing and analysing complex relationships, graph databases prove to be the front-runner in the industry for various software and solutions (C5) (Stanescu, 2021).

3.3. Relational databases

Relational databases are widely used for data management in both industry and academia (C4, C5) (Lazarska & Siedlecka-Lamch, 2019). This is due to their efficient data storage and retrieval capabilities (Ravat et al., 2020). Although they can handle structured data queries well (C2) (Ramis Ferrer et al., 2021), relational databases face challenges in dynamic visualisation (C1) due to their tabular structure and reliance on complex joins (Lazarska & Siedlecka-Lamch, 2019). Moreover, visualisation tools may struggle with real-time interactive representations (C1). Relational databases excel in querying with Structured Query Language (SQL), allowing efficient data retrieval and manipulation (C2). However, they lack depth in semantic representation (C3), focusing more on data storage and retrieval rather than complex semantic relationships (Martinez-Cruz et al., 2012). Ultimately, of the three different representations, relational databases seem less suitable for the proposed dynamic database.

3.4. Synthesis

Drawing on the description of each potential representation and the design criteria set out in Table 1, graph databases appear to be the preferred data representation model for the proposed database, given their relatively strong performance across multiple criteria. The main driving force for this decision is the dynamic visualisation this representation provides, allowing for interactive and real-time navigation (Figueres-Esteban et al., 2018). This will enable users to study and analyse links between research papers. Graph databases are also sufficient in terms of scalability, speed, efficiency and flexibility. They

provide a satisfactory output in this regard due to them being well-suited for interconnected data. Whilst ontologies are preferred for semantic representation, graph databases are preferred as the proposed database focuses more on functionality rather than exclusively on semantics. Indeed, the implementation provided by graph databases appears to be sufficient given the scoped semantic needs in the initial maps from Guérineau et al. (2022).

4. Design and functionality of the proposed Dynamic Database

Based on the selected representation of graph databases, the proposed community-driven database for the dynamic representation, "Dynamic Database" in short, is detailed in this section. Design-wise, its architecture and its constituents need to be defined to reflect the ones from the initial maps. To this end, the next section presents two UML diagrams, before delving into offered functionalities.

4.1. UML diagrams: representing the Dynamic Database architecture

The initial maps from Guérineau et al. (2022) depict a collection of blocks representing concepts or techniques, often interconnected with links denoting associations between them. The blocks and the links are supported by references identified from the scientific literature. To transfer this formalism, a Class Diagram UML model is presented hereafter in Figure 1. A second UML model, a Use Case diagram, defines the different usage and expected functionalities. These UML models elaborate upon the structure of the proposed Dynamic Database. These two diagrams can provide an ease of comprehension regarding what the database entails, and why it may be needed.

Figure 1. UML Class Diagram of the Dynamic Database

The Class Diagram, shown in Figure 1, illustrates the structural elements of the Dynamic Database, modelled according to UML standards. It includes several key classes – Map, Block, Reference, and Link – which form the core building blocks of the database. These classes are supplemented by four enumerations: RType, LType, MapLevel, and CType. Each of these four enumerations representing different types of relationships or categorisations from the initial maps.

The Map class represents the overarching container for the entire map representation, functioning as a central point for managing both blocks and references. Its main attributes include a list of blocks, stored in the BlockList attribute, and a list of references, stored in the ReferenceList. The Map class is equipped with functions to display, filter, and update the map, ensuring dynamic visualisation and interaction.

Moving on to the Block class, this represents the core unit within the map – each block corresponds to an individual concept or technique. Blocks can be connected to other blocks, reflecting the relationships between different concepts and techniques, and these connections are managed through the Link and Reference classes. The Block class supports several functions, including the ability to add, update, and delete blocks, making it adaptable for real-time modifications in the Dynamic Database. The Reference class, on the other hand, manages the bibliographic elements tied to these blocks, including attributes such as "title", "author", and "year". It similarly allows for references to be added or updated as needed. The Link class plays a vital role in connecting the blocks, representing relationships through a triplet structure, namely "source", "reference", "target". This allows for advanced querying and filtering features in the database. The UML links in the Class Diagram ensure structure and flexibility by establishing clear relationships between the Link, Reference, and Block classes. The Link class connects blocks via references, while the Block class organises and categorises these connections into a navigable map. The Reference class provides context for the connections, allowing for the integration of relevant sources. This allows support for a structured system that can dynamically update and adapt, making it easier to navigate and change map content.

These classes and their relationships ensure that the Dynamic Database reflects the initial maps' formalism and semantics while allowing it to evolve and scale. By utilising enumerations such as RType and LType, the database ensures consistency in how links and references are categorised, making it easier for users to filter maps. The following paragraphs elaborate on these features.

The Use Case diagram, on Figure 2, highlights the interaction between two primary actors, namely the User and the Peer Reviewer. The Peer Reviewer inherits all use cases from the User, as depicted by the arrow between them. This means that the Peer Reviewer shall have all the capabilities mentioned in for the User, with the added capability of reviewing changes made to the existing representation.
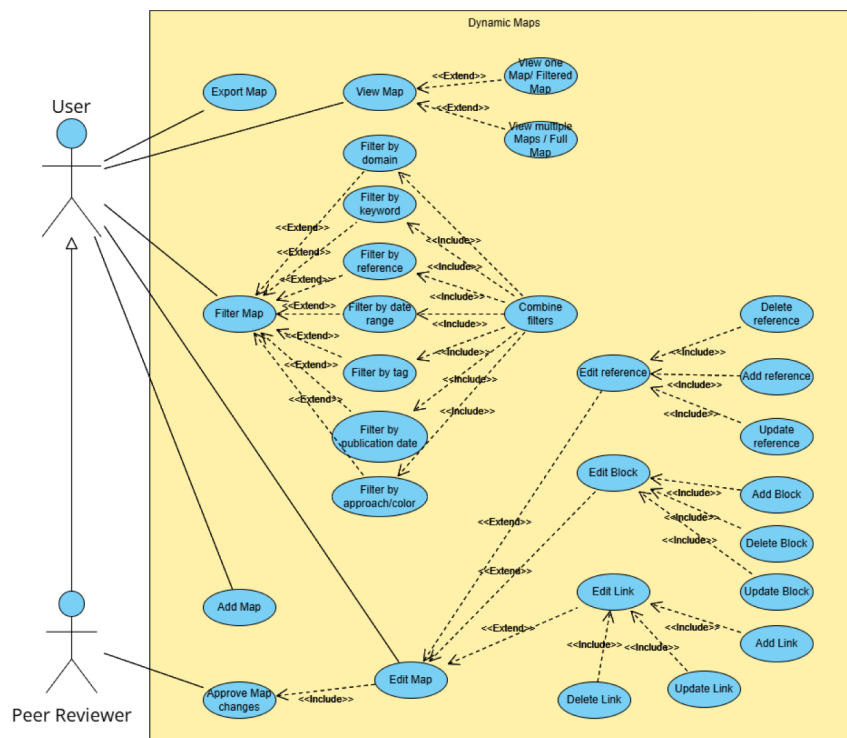

Figure 2. Use Case diagram for the Dynamic Database

The Dynamic Database offers various functionalities related to the management of dynamic maps, which are grouped under functions like filtering, editing, and map management. The User interacts with the database primarily through functions such as viewing, filtering, and editing maps. The ability to filter the maps by different criteria – such as domain, keyword, or publication date – allows users to tailor the visualisation according to their needs. This flexibility is extended through combinations of filters, giving users the possibility to refine their queries and focus on specific subsets of the map.

On the other hand, the Peer Reviewer engages with the database from a different perspective, primarily focusing on the approval and validation of changes made by the user. This includes approving updates to

maps, references, blocks, and links. The Peer Reviewer ensures that any changes made to the Dynamic Database align with predefined standards and contribute positively to the repository. Both actors benefit from the database's dynamic architecture, which supports real-time updates and visualisations. The Use Case diagram not only shows how these interactions occur but also clarifies the extensibility of the database, indicating that new functionalities can be added as the database evolves. The diagram effectively communicates the processes by which maps can be manipulated, filtered, and refined, emphasising the database's capacity to adapt to the needs of its users while maintaining academic rigour. By presenting these UML diagrams, readers can better understand how the database is structured – through the Class Diagram – and how it operates – through the Use Case diagram. Each figure complements the other, providing a comprehensive view of both the database's architecture and its user interactions. The next section expands on the core functionalities in line with the UML design.

4.2. Core functionalities of the Dynamic Database

As represented on Figure 2, the core functionalities of the database aim to provide a filterable and modifiable representation that fetches data in real-time and updates according to the User's inputs. These functionalities propose a solution to satisfy the design criteria and, by extension, the main objectives of the research project. The View, Filter and Edit functionalities are described in the following paragraphs. The View Map use case (Figure 2) outlines the ability of the database to correctly visualise all concepts and techniques, grouping them into their correct level (i.e., approach, process, method or tool), with their associated links and references. A side-by-side comparison of how the initial representation compares to the database's visualisation is shown in Figure 3. The emphasis is on the lossless association of information from the initial representation to the dynamic representation.

Figure 3. Comparison of initial representation in Guérineau et al. (2022) (A) with the proposed representation in the developed Dynamic Database interface (B)

The Filter Map use case outlines the filtration of the concepts and techniques, their links and associated references inside the selected map (see Figure 4). Unlike taxonomy-based classification systems, which rely on predefined hierarchical categories and top-down filtering mechanism, the proposed one allows the User to select multiple criteria and customise their searches. This functionality enables users from both academia and industry to express their idea more thoroughly without the need to manually scan through every single concept or technique given. Instead, one can utilise the filtration capabilities of the database to extract a subset of concepts and techniques. Said capabilities are shown in Figure 4, allowing for a user-friendly way to filter the maps. The Filter by Author/Reference option allows selection of specific authors or references, while the Filter by Tag enables narrowing down by community-defined topics, e.g., analysis, or system architecture. Filter by Color visually distinguishes between different product development approaches (Systems Engineering, Agile, agnostic, etc.). The Filter by Year and by Keyword options further refine results by time and specific terms extracted from the papers. The different

filters can be combined, allowing the users to narrow down their search, navigate and export the resulting maps. Finally, the Navigation Bar, at the top of Figure 4, in light red, offers the possibility to add, delete, and modify blocks and references, which are key functionalities to the Edit Map use case discussed in the next paragraph.

Figure 4. Screen shot of the Dynamic Database showing all filtration options available

The Edit Map use case (Figure 2) outlines the intent to create an updateable repository of concepts and techniques, in line with the second main objective. To reflect the evolution of the domain and the new literature pertaining to MPD, the database needs to be maintained and sustained for the long-term benefit of academia and industry. In this regard, a peer-reviewing system that would allow Users of the database to propose additions and modifications to concepts and techniques, their links and references is proposed. These changes will then go through a "review" process. The purpose of this review is to ensure that all references are valid and correctly cited, and that the proposed changes are consistent with the original content of the referenced papers, accurately reflecting the concepts and techniques discussed, and the associated links. Such a review process is intended to minimise conflicting contributions. This review process will involve a Peer Reviewer (Figure 2), for instance, an "expert" user such as a researcher with the relevant expertise. Said Peer Reviewer will have access to this set of reviewable blocks and references, and will have the additional capability of approving or denying such changes. The Peer Reviewer will also be able to modify and delete existing blocks and references to eliminate any discrepancies they may find. To ensure only trusted users are provided with such sensitive permissions, and to avoid any unnecessary changes to the representation, user authentication is divided into two separate streams. A first stream, relying on a Firebase user authentication, will be for the general public, allowing them to create an account. Such Users shall gain access to display, filter and export the map. However, these Users may only be allowed to propose changes to the existing maps with no power to modify the real-time representation in the repository. Accordingly, these proposed changes will not be visible in the current real-time representation of the map and will be temporarily stored until verified by a Peer Reviewer. The latter is identified thanks to an ORCID account that will be used to login via the second stream. Any approved changes by the Peer Reviewer will be committed into the database, and the changes will be visible and live to every other user that accesses the database. This set of functionalities provides a database that can benefit the research community, and aid in the industry.

5. Discussion and perspectives

Previous sections presented the proposed Dynamic Database through its architecture, its uses cases and core functionalities. This section discusses the positioning relative to existing repositories, as well as the design choices and implications for research and industry, before addressing the opportunities for improvement.

5.1. Relation to other repositories and initiatives

The Dynamic Database differs from repositories discussed by Mayookh & Srinivasan (2023). Notably, it is both community-driven and web-based, whilst also integrating peer-review and multi-criteria filtering mechanisms. Furthermore, it aims to cover a broader range of concepts and techniques, organised from approach to tool level, with a focus on MPD. In contrast, the identified repositories tend to focus on

specialised design methods, such as human-centred design methods or usability (Kramer et al., 2017; Mayookh & Srinivasan, 2023). Another distinctive feature is the emphasis on the graphical representation of concepts and techniques and their links from peer-reviewed papers from the scientific literature. In this respect, and to the authors' knowledge, this is an original direction. However, some of the limitations highlighted by Gericke et al. (2016) remain unaddressed at this stage. Complementarily, other features included in the identified repositories such as the integration of empirical data, support for the application of concepts and techniques, or glossaries, also stand out as future opportunities.

5.2. Design choices: selecting graph database over ontology

Whilst the comparison tends to favour the performance of graph databases for this project, the opportunity cost of preferring said representation over ontology needs to be discussed as well. The decision to prioritise query performance, dynamic representation, and scalability by choosing a graph database comes at the cost of the advanced semantic reasoning and in-depth knowledge representation that ontologies can offer. This trade-off may impact the richness of data representation, which could be a concern for certain research applications. However, the scalability and flexibility of graph databases make them better suited for handling large volumes of data. In future work, the integration of ontologies into the Dynamic Database (Ben Mahria et al., 2021) could create a hybrid solution that combines the semantic insights of ontologies with the powerful capabilities of graph databases, addressing the limitations of both representations. Such an integration could lead to a more complete solution, benefiting both research and industry by providing rich semantic data along with enhanced performance and scalability.

5.3. A research perspective: importance of the researchers' contribution

From a research perspective, the proposed Dynamic Database can be envisioned as a proof of concept for graphically organising and sharing knowledge on concepts and techniques and how they can be linked. These links are manually extracted from the research papers. Therefore, the role of researchers in contributing, maintaining and evolving the proposed Dynamic Database is critical for its long-term viability and remains a major limitation to the proposed work. Indeed, the Dynamic Database requires the collaboration and willingness of researchers to adopt this database, and contribute to it. In turn, this up-to-date database can be utilised by the same community as a collaborative database for searching, filtering, analysing and exporting query results. The latter could help the research community to identify research gaps, trends or consensus on the use of concepts and techniques. Concurrently, further work is required to investigate the researchers' willingness to adopt it, and collaborate around it, which could be done through workshops and focus groups (Caillaud et al., 2016). This collaboration could eventually be consolidated through institutional partnerships and endorsed by a community of practice. A similar investigation should be concurrently considered with the industry, which might have different interests for the project, as discussed below.

5.4. An industry perspective: the selection mechanism

Part of the interest for the industry to use the Dynamic Database would be to support them in the selection of the concepts and techniques to be used for their MPD. The proposed database currently relies on the user's expertise to that extent. In that sense, a potential improvement would be to implement one of several selection mechanisms to elevate the usability. For instance, the selection mechanisms through designers' needs (López-Mesa & Thompson, 2003), or through project and organisational context (Hollauer et al., 2017) could be implemented in future work. For the moment, an alternative solution is using the tag-based filtration.

5.5. Limitations and future steps

In addition to the points discussed above, several key steps are considered for future work. From a technical standpoint, over time, as the volume of data in the repository is expected to grow, ensuring horizontal scalability will be crucial. To address horizontal scalability as the database grows, future work should focus on optimising query performance and expanding database architecture.

The UML class and use case diagrams outlined the technical design of the Dynamic Database and its core functionalities, reflecting the initial vision of the research team. Future steps include evaluation and design iterations in collaboration with future users. For instance, the user-interface (UI) and the user-

experience (UX) remain limited at this stage, and their improvement will certainly contribute to a better usability and adoption from both research and industry. Users' feedback can be obtained through pilot testing and focus groups, ensuring that the project evolves to meet the needs of its intended users. Finally, the integration of artificial intelligence is a natural future step to enhance the filtration.

6. Conclusion

The project is a preliminary step towards establishing a foundation for a community-driven database to represent concepts and techniques for MPD. First, three data representations, namely ontologies, graph databases and relational databases are compared. This comparison is supported by criteria derived from the sub-objectives, which in turn support the two main objectives of the project. The comparison highlighted the suitability of graph databases for the project criteria. To illustrate this, a UML Class Diagram and a Use Case diagram are presented to outline the architecture and core functionalities of the Dynamic Database. These include features such as viewing, filtering, and updating maps to enhance data exploration and analysis, thereby supporting further insights. Some of these functionalities are illustrated through various screenshots of the Dynamic Database in parallel with its original representation, showing improvements in usability and enhanced filtering capabilities. These visual examples serve as a proof of concept, demonstrating the feasibility and potential of the community-driven database. In line with this community aspect, the importance of researchers in participating in and maintaining the database is emphasised. Indeed, their contributions, through peer-reviewed additions, will help the database to follow the evolution of the research work proposed by the community, enabling a greater impact on both research and industry.

Acknowledgments

Supplementary material: The code of the proposed Dynamic Database, its associated technical documentation and a demonstration video are publicly available on GitHub (Bilal & Guérineau, 2025).

References

Baheti, R., & Gill, H. (2011). Cyber-physical Systems. *The Impact of Control Technology*, 1, 161–166.

Ben Mahria, B., Chaker, I., & Zahi, A. (2021). A novel approach for learning ontology from relational database: from the construction to the evaluation. *Journal of Big Data*, 8(1). https://doi.org/10.1186/s40537-021-00412-2

Bilal, M., & Guérineau, J. (2025). *Product Development Maps*. Zenodo - Github. https://doi.org/10.5281/zenodo.14993432

Böhmer, A. I. (2018). *When digital meets physical – Agile Innovation of Mechatronic Systems*, Doctoral thesis, Technische Universität München.

Bradley, D., & Hehenberger, P. (2016). Mechatronic futures. In Hehenberger, P., Bradley, D. (Eds.) *Mechatronic Futures* (pp. 1–15). Springer. https://doi.org/10.1007/978-3-319-32156-1_1

Caillaud, E., Rose, B., & Goepp, V. (2016). Research methodology for systems engineering: some recommendations. *IFAC-PapersOnLine*, 49(12), 1567–1572. https://doi.org/10.1016/j.ifacol.2016.07.803

Chen, Y., & Xing, X. (2022). Constructing Dynamic Knowledge Graph Based on Ontology Modeling and Neo4j Graph Database. *2022 IEEE 5th International Conference on Artificial Intelligence and Big Data, ICAIBD 2022* (pp. 522–525). https://doi.org/10.1109/ICAIBD55127.2022.9820199

Figueres-Esteban, M., Hughes, P., El Rashidy, R. A. H., & van Gulijk, C. (2018). Manifestation of ontologies in graph databases for big data risk analysis. *Safety and Reliability - Safe Societies in a Changing World - Proceedings of the 28th International European Safety and Reliability Conference, ESREL 2018* (pp. 3189–3194). https://doi.org/10.1201/9781351174664-399

Gericke, K., Kramer, J., & Roschuni, C. (2016). An exploratory study of the discovery and selection of design methods in practice. *Journal of Mechanical Design*, 138(10), 101109. https://doi.org/10.1115/1.4034088

Goevert, K., & Lindemann, U. (2018). Further development of an Agile technique toolbox for mechatronic product development. In Marjanović D., Štorga M., Škec S., Bojčetić N. (Eds.), *Proceedings of International Design Conference, Volume 5: DESIGN2018* (pp. 2015–2026). https://doi.org/10.21278/idc.2018.0204

Guérineau, B., Rivest, L., Bricogne, M., Durupt, A., & Eynard, B. (2018). Towards a Design-Method Selection Framework for Multidisciplinary Product Development. In Marjanović D., Štorga M., Škec S., Bojčetić N.

(Eds.), *Proceedings of International Design Conference, Volume 5: DESIGN2018* (pp. 2879–2890). https://doi.org/10.21278/idc.2018.0431

Guérineau, J., Bricogne, M., Rivest, L., & Durupt, A. (2022). Organizing the fragmented landscape of multidisciplinary product development: a mapping of approaches, processes, methods and tools from the scientific literature. *Research in Engineering Design*, 33(3), 307–349. https://doi.org/10.1007/s00163-022-00389-w

Hollauer, C., Becerril, L., Kattner, N., Weidmann, D., Chucholowski, N., & Lindemann, U. (2017). Adaptable mechatronic engineering design processes: Process reference model and methodology. In Chakrabarti, A., Chakrabarti, D. (Eds.) *Research into Design for Communities, Volume 1. ICoRD 2017. Smart Innovation, Systems and Technologies*, vol 65 (pp. 597–607). Springer. https://doi.org/10.1007/978-981-10-3518-0_52

Kortuem, G., Alford, D., Ball, L., Busby, J., Davies, N., Efstratiou, C., Finney, J., White, M. I., & Kinder, K. (2007). Sensor Networks or Smart Artifacts? An Exploration of Organizational Issues of an Industrial Health and Safety Monitoring System. In Krumm, J., Abowd, G.D., Seneviratne, A., Strang, T. (Eds.) *UbiComp 2007: Ubiquitous Computing. UbiComp 2007. Lecture Notes in Computer Science*, vol 4717 (pp. 465–482). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-74853-3_27

Kramer, J., Poreh, D., & Agogino, A. (2017). Using TheDesignExchange as a knowledge platform for human-centered design-driven global development. *Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 1: Resource-Sensitive Design | Design Research Applications and Case Studies*, Vancouver, Canada (pp. 101–110). www.thedesignexchange.org

Lazarska, M., & Siedlecka-Lamch, O. (2019). Comparative study of relational and graph databases. *2019 IEEE 15th International Scientific Conference on Informatics* (pp. 363–370). https://doi.org/10.1109/Informatics47936.2019.9119303

López-Mesa, B., & Thompson, G. (2003). Exploring the need for an interactive software tool for the appropriate selection of design methods. In Folkeson A., Gralen K., Norell M., Sellgrem U. (Eds.), *DS 31: Proceedings of ICED 03, the 14th International Conference on Engineering Design* (pp. 1–10).

Martinez-Cruz, C., Blanco, I. J., & Vila, M. A. (2012). Ontologies versus relational databases: Are they so different? A comparison. *Artificial Intelligence Review*, 38(4), 271–290. https://doi.org/10.1007/s10462-011-9251-9

Mayookh, M., & Srinivasan, V. (2023). A Review of Repositories of Design Methods. In Chakrabarti A., & Singh, V. (Eds.) *Design in the Era of Industry 4.0, Volume 1. ICORD 2023. Smart Innovation, Systems and Technologies*, vol 343 (pp. 1205–1215). Springer, Singapore. https://doi.org/10.1007/978-981-99-0293-4_97

Miranda, J., Pérez-Rodríguez, R., Borja, V., Wright, P. K., & Molina, A. (2017). Sensing, smart and sustainable product development (S3product) reference framework. *International Journal of Production Research*, 7543, 1–22. https://doi.org/10.1080/00207543.2017.1401237

Pokorný, J. (2015). Graph Databases: Their Power and Limitations. In Saeed K., & Homenda W. (Eds.) *Computer Information Systems and Industrial Management. CISIM 2015. Lecture Notes in Computer Science*, vol 9339 (pp. 58–69). Springer, Cham. https://doi.org/10.1007/978-3-319-24369-6_5

Ramis Ferrer, B., Mohammed, W. M., Ahmad, M., Iarovyi, S., Zhang, J., Harrison, R., & Martinez Lastra, J. L. (2021). Comparing ontologies and databases: a critical review of lifecycle engineering models in manufacturing. *Knowledge and Information Systems*, 63(6), 1271–1304. https://doi.org/10.1007/s10115-021-01558-4

Ravat, F., Song, J., Teste, O., Trojahn, C., & Santos, D. (2020). Efficient querying of multidimensional RDF data with aggregates: Comparing No SQL, RDF and relational data stores. *International Journal of Information Management*, 54. https://doi.org/10.1016/j.ijinfomgt.2020.102089

Sir, M., Bradac, Z., & Fiedler, P. (2015). Ontology versus Database. *IFAC-PapersOnLine*, 28(4), 220–225. https://doi.org/10.1016/j.ifacol.2015.07.036

Stanescu, L. (2021). A Comparison between a Relational and a Graph Database in the Context of a Recommendation System. In Ganzha M., Maciaszek L., M. Paprzycki M., & Ślęzak D. (Eds.) *Position and Communication Papers of the 16th Conference on Computer Science and Intelligence Systems*, 26, 133–139. https://doi.org/10.15439/2021f33

Zhang, X., & Thomson, V. (2016). The impact and mitigation of complexity during product design. *International Journal of Design and Nature and Ecodynamics*, 11(4), 553–562. https://doi.org/10.2495/DNE-V11-N4-553-562