Empowering Worker-Robot Collaboration: Leveraging LLMs for Extracting and Visualizing Robot Task Metrics

Luigi Gargioni[1★][0000-0003-4354-916X] and Daniela Fogli[1][0000-0003-1479-2240]

[1]Department of Information Engineering, University of Brescia, Brescia, Italy
luigi.gargioni@unibs.it, daniela.fogli@unibs.it

Abstract. In the context of Industry 5.0, which is characterized by a close integration between digital technology, industrial production, and human-centered design, collaborative robots emerge as key players. These robots are no longer isolated machines but an integral part of an interconnected ecosystem, where the fluidity of data plays a crucial role. Collaborative robots facilitate flexibility, efficiency, and safety in operations. However, this also introduces novel programming and data management challenges. A distinctive feature of collaborative robots is their ability to be programmed and used by non-expert users. This democratization of access to robotics offers significant advantages but also requires careful design of tools and interfaces to enable easy access to the data generated by the robots. In this context, the user interface assumes a pivotal role in ensuring that even those lacking programming expertise can fully benefit from the capabilities of collaborative robots and the data they produce. This study exploits Human Work Interaction Design principles to examine the problems encountered by individuals lacking programming skills when attempting to obtain data about robot performance. A solution that exploits Large Language Models is proposed.

Keywords: Meta-Design · Large Language Model · Human-Robot Interaction · Collaborative Robot · Data Visualization

1 Introduction

Data has become a valuable resource in the digital era, driving innovation and guiding strategic decisions across various sectors. Among these data, those generated by the use of collaborative robots can provide important information for decision-making, thus entailing a significant impact on human work and well-being. Collaborative robots are robots that can stay in close contact with humans thanks to technologies and sensors designed to ensure the physical safety of the operator [24]. They are usually employed beyond the traditional industrial production lines, to carry out tasks in close collaboration with human operators [21]. Collaborative robotics is a research field with significant potential impacts

★ Corresponding author

on industries, services, and societies at large, fostering an economic development model that is sustainable and respectful of the environment and people [19]. Indeed, collaborative robotics is a central element of Industry 5.0 [5], which in turn pays attention to creating collaborative, interconnected, and adaptable industrial settings, where a meaningful interaction between workers and technology contributes to achieving higher efficiency, effectiveness, and customization of manufacturing and production processes [11, 26]. Industry 5.0 strongly emphasizes sustainable workplaces aiming to optimize processes and consumption, reduce waste materials, and improve human safety and well-being at work [22, 2]. Collaborative robotics represents one of the technologies that can contribute to achieving these goals by supporting a safe collaboration between robots and humans and ensuring that workers' expertise is properly valued in production processes.

The literature reports that there are various types of data that may be of interest when performing tasks with collaborative robots. The paper [18] underlines the importance of productivity-related metrics, such as the average time taken to complete a specific category of tasks and the relative completion frequency. In [7], the efficiency of robot usage also comes into play in terms of the robot's up-time compared to the total available time. This is also an important indicator in terms of the economic viability of the investment in an automated solution. Another interesting factor, as reported in [14], is resource usage: this refers to monitoring the usage of tools and materials required during task execution to obtain an indicator for predictive maintenance. Another factor, probably the most important, is work safety, which includes monitoring of safety issues while interacting with the robot (e.g., stop due to collision) [13, 18].

Usually, data of interest for robot users are represented in dashboards that aggregate them in different ways. The problem is that these dashboards statically show only the data and the visualization that the developer decided to include when creating it. Therefore, accessing and analyzing the data generated by collaborative robots remain challenging activities, often reserved for technical experts and industry specialists. Different workers with different roles in the company may need different data aggregated or presented differently. It becomes necessary for each worker to obtain customized data visualization simply by requesting the data and the visualization format they prefer, without the developer having to provide this combination at design time.

The application of Human Work Interaction Design (HWID) methods to work analysis led us to propose a natural language-based approach to democratize access and analysis of robot usage data. This approach would make data available and understandable even for people without specific technical training, facilitating the engagement of a broader base of interested users. In particular, the adoption of Large Language Models (LLMs) to obtain and analyze robot usage metrics will be explored in this paper. It will focus on the importance of extracting and analyzing usage data in a simple and accessible manner for end users, regardless of technical expertise.

The paper is organized as follows. Section 2 provides an overview of the related works; Section 3 details the user-system interaction assumed for the proposed solution; Section 4 presents the technical implementation of the proposed solution with two different approaches; Section 5 describes a use case to show how the different solutions would be applied; and, finally, Section 6 provides concluding remarks and discusses future works.

2 Related Work

The capability of LLM systems to comprehend natural language and discern users' intentions has prompted an investigation into their potential applications in various domains. Recent proposals have been put forth to use LLMs to enhance the easiness and naturalness of web development, such as generating mock-ups of web interfaces [10] or snippets of HTML, Javascript, and CSS [12], on the basis of textual descriptions. To address potential shortcomings of LLMs, such as generating unnecessary or inaccurate content, these tools provide end users with several controls: 1) displaying multiple output options, 2) allowing editing of the model's output, and 3) facilitating transitions to external resources via web searches. As a further advancement, the prompt engineering technique explored in [3] enables end users to refine LLM outputs through iterative input refinements, thereby allowing them greater influence over the generated web pages. However, users must be familiar with web terminology to effectively convey their design intentions to the LLM. A similar approach has been experimented in the field of robot programming, as highlighted by the system described in [23], which uses OpenAI ChatGPT. Here, users are tasked with providing feedback on the quality and safety of Python or C++ code generated by ChatGPT. However, this requires that users comprehend the generated code and are capable of suggesting appropriate corrections, thereby requiring programming proficiency. In an attempt to address this challenge, the approach described in [6, 8] explores the use of a graphical visualization of the program generated by a Natural Language Processing (NLP) interface. This interface also enables users to directly manipulate the program through its graphical representation.

As far as the problem considered in this paper, namely, fostering access and analysis of robot-generated data by workers without programming competencies, natural language understanding might be exploited to construct Structured Query Language (SQL) queries. The automatic generation of SQL queries is not a novel concept. For instance, in [17], a rule-based NLP approach is employed to interpret the query description provided by the user. With the advent of LLMs, a series of analyses have been conducted to evaluate the effectiveness of the SQL queries produced in accordance with the user request [25] and their efficiency during execution [20]. Thus far, the outcomes have been promising but still far from an optimal solution [16]. Furthermore, additional research is required to facilitate end users' usage of data.

This paper proposes the adoption of HWID in the design of an LLM-based application that can address the aforementioned need, considering the data generated by collaborative robots as a case study. As previously underlined, these robots are designed for use in environments beyond the traditional factory setting. Consequently, the retrieving of data and metrics associated with the robot tasks should be performed by workers who lack programming expertise. Interrogation of databases should, therefore, be made as simple as possible, potentially using natural language.

3 Interaction Design

HWID promotes the integration of work analysis and human-centered interaction design to develop better workplaces and sustain improved work practices [4, 1]. Following the methods and principles of HWID, we studied how to make dashboards related to the use of collaborative robots more flexible and suitable for different human workers.

We observed and interviewed healthcare professionals in the frame of a project about supporting the preparation of personalized medicines with cobots [9]; on that occasion, we gathered workers' needs related to data visualization and exploration tasks that require personalized dashboards. Specifically, we developed healthcare worker personas to better frame their goals and frustrations. In particular, we found that in the healthcare sector, current technologies are often imposed by manufacturers without taking into account the needs of the end users. As a result, it is the user who has to adapt to the proposed functionalities rather than the tools that are tailored to real workers' needs. The same argument can be applied to data visualization features provided in commercial dashboards.

A conventional dashboard typically displays the most important metrics, commonly called Key Performance Indicators (KPIs). However, guessing all the information that any user might require is challenging. Furthermore, data of interest may change over time, even partially (e.g., the time interval), and users who utilize these tools may also change. The proposed interaction is designed to facilitate the extraction of information, avoiding the necessity for a developer to predict it within a pre-defined dashboard.

In accordance with the HWID principles, the initial stage of the interaction design process entails the development of preliminary mock-ups. The user interface mock-up in Figure 1 presents a possible interaction scenario between the user and the system. The interface includes a chat panel on the left and a data visualization panel on the right. In the chat panel, the user can express their requests by writing or even speaking. In the considered scenario, the user needs to obtain the robot usage time in the current month in comparison with the previous month. This statistic could be useful to determine the robot's up-time compared to the total available time. If the user does not express any specific preference for data visualization, the assistant will take care of this aspect based on the results obtained from the query executed on the database. Once the extracted data has been displayed using the chosen or suggested visualization type, it can be further refined in terms of both the data itself, namely editing the proposed query, and the representation type. Figure 2 illustrates an example where the user requests a change of visualization and a different comparison metric.

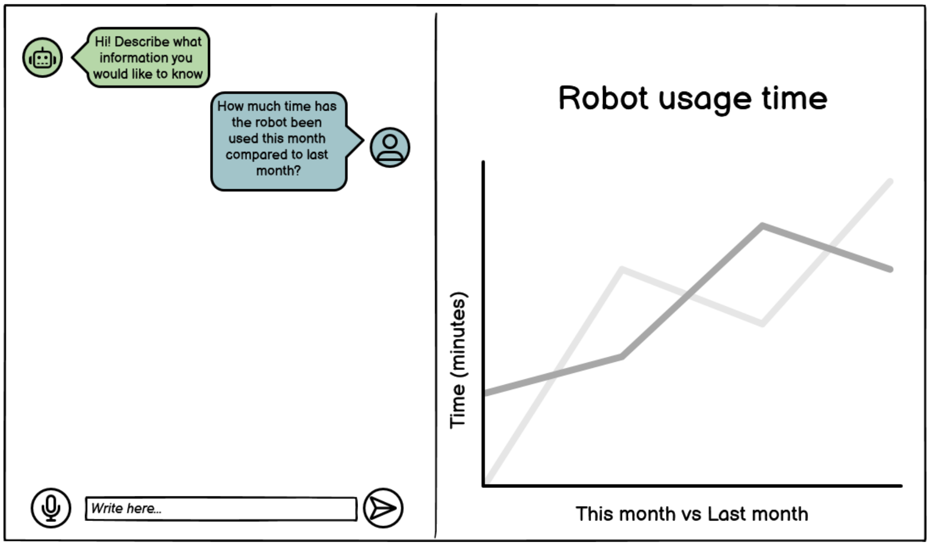

Fig. 1. The interface mockup

4 The Implemented LLM-based Solution

The proposed solution is based on a web application that employs the client-server paradigm. Specifically, the database used is PostgreSQL, and therefore the query examples presented later will adhere to PostgreSQL syntax. To exploit the functionalities of an LLM, the APIs provided by the LLM service provider will be used. In this context, OpenAI ChatGPT offers an easily integrated API service and a model that is currently one of the most powerful (*GPT-4o*).

The solution can be implemented in two ways: with a workflow that includes two model instances or with a workflow that includes only one model instance The main difference is whether to have a single model instance or two separate model instances to i) create the extraction query based on the user intent and ii) suggest the most appropriate visualization for the extracted data.

The two approaches are outlined below, and a subsequent analysis will be presented, evaluating the advantages and disadvantages of each of them.

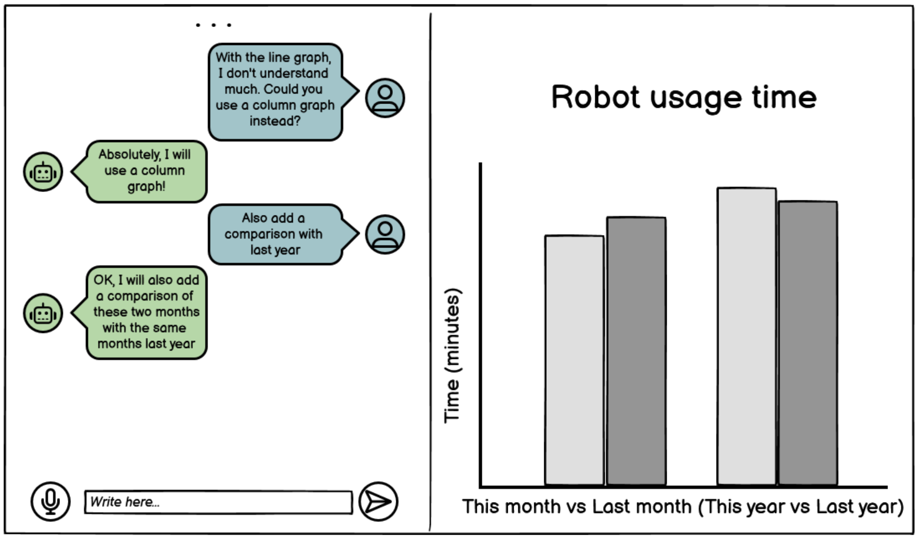

Fig. 2. The refinement interaction

4.1 Interaction Workflow Based on Two Model Instances

In this workflow (Figure 3), two distinct instances of the model are used, each one appropriately instructed:

- *First instance*: to recognize the user request and produce the SQL query to access the database;
- *Second instance*: to suggest the recommended data visualization based on the extracted data.

To instruct the first instance of the model, which is responsible for query creation, it is necessary to provide it with a specific instruction set. These directives will inform the model about the context and the specific task. The instructing process, namely *prompt engineering*, is conducted in natural language by the developer of the application and includes phrases such as *"You are an assistant who must help the staff of a laboratory to extract information from a database about their robots and their tasks. If the user's request is not clear, ask for more details before proposing an SQL query. Return an SQL query that can be executed on this database schema."*. In addition to contextual information, it is essential to provide the model with the structure of the database to enable it to propose formally correct queries. Furthermore, the inclusion of some example queries among the most common ones that might be asked helps the model considerably in its task. The latter will not constrain the model to respond only to those queries, but rather assist it in developing a more nuanced understanding of how to respond. Before execution on the database, the query proposed by the model is subjected to a formal syntax check to ensure that it adheres to

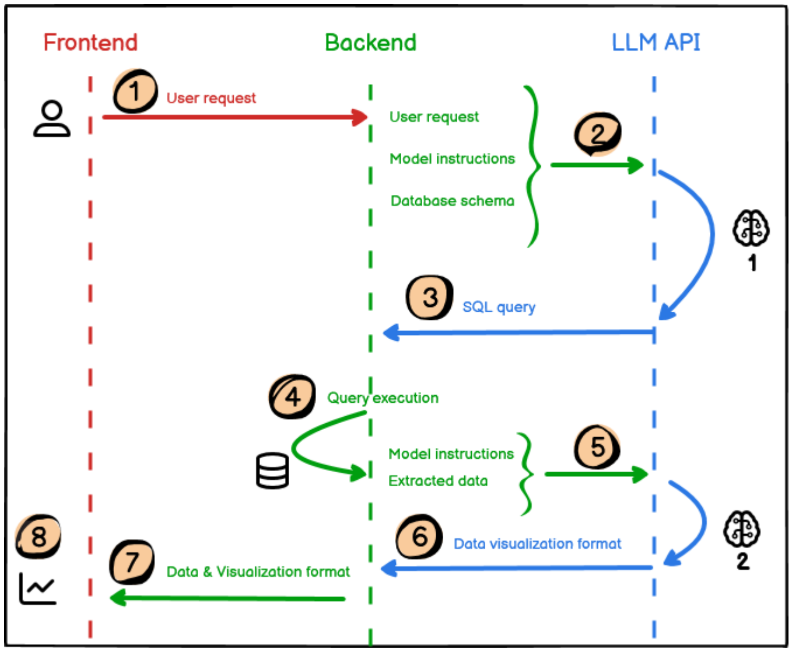

Fig. 3. The workflow based on two model instances

the correct SQL syntax and that it only contains data selection operations (*SELECT* statements), and not edit or delete operations (*UPDATE* or *DELETE* statements), thus preventing any potential security breaches.

In contrast, for the initialization of the second model instance, the one designed to suggest a proper data visualization based on the retrieved data, it is sufficient to provide contextual information such as *"Based on the data the user will provide, reply with how they might be represented so that they are usable and understandable in the best way possible."*.

For both instances, it is crucial to minimize the parameter that regulates the variation of the responses to the same input, called *temperature*, to obtain answers that are as deterministic as possible.

Figure 3 shows the workflow during user interaction, also making the parts of the architecture involved explicit. The process starts with the user's request (*Step 1*) in natural language, as illustrated in Figure 1. This is sent to the back-end server and then sent, along with the model initialization instructions, the history of the previous messages in this conversation, and the database structure, to the LLM (*Step 2*) through its API. The first instance of the model will interpret the user's request and, given the database structure, will reply with a specific SQL query to extract the requested information (*Step 3*). The query is then executed on the database (*Step 4*), and the data are obtained. An additional request containing the extracted data (*Step 5*) is sent to the LLM using the second model instance to retrieve the recommendation of the most appropriate data representation (*Step 6*). Then, the data and the recommended representation are then sent to the front-end (*Step 7*) for the final visualization (*Step 8*). Depending on the result, the user may refine their request to correct or improve the information obtained. This can be easily done through a conversational interaction as illustrated in Figure 2.

4.2 Interaction Workflow Based on One Model Instance

Figure 4 reports the workflow based on a single model instance. It is quite similar to the one based on two model instances: the only differences are in *Step 3*, where the model provides both the SQL query and the data visualization format to the backend devoted to query execution, and in *Step 5*, where the data are passed directly to the frontend to be visualized, rather than to the second model instance. The model instance prompt is also quite similar to the prompt of the first model instance in the other workflow. This prompt has been modified simply by adding an instruction to provide the data visualization format suggestion in addition to the SQL query, relying only on the user's request. All the rest of the parameters and supporting functions are the same.

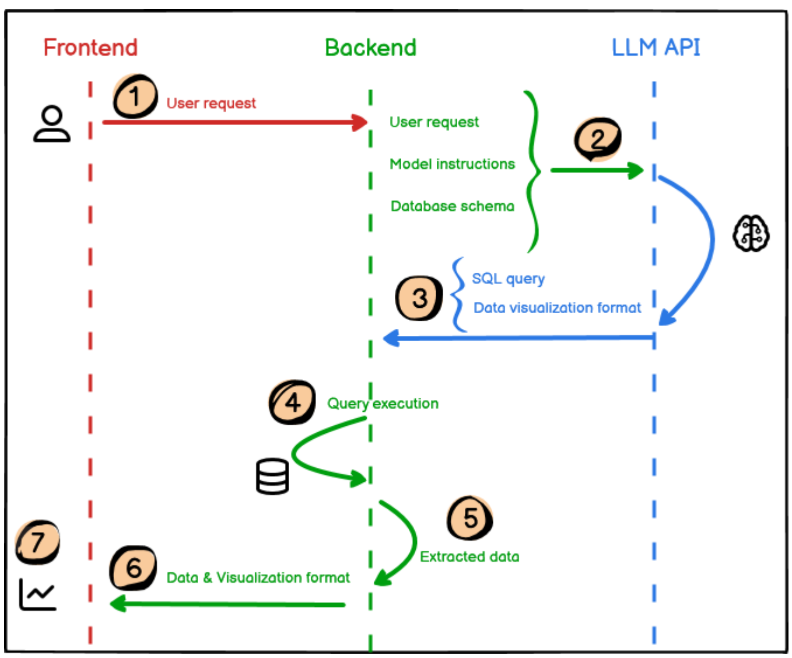

Fig. 4. Alternative workflow based on a single model instance

4.3 Comparing the Two Interaction Workflows

Both approaches have advantages and disadvantages that should be taken into consideration to make the best choice based on the situation at hand.

The workflow comprising two model instances offers an effective representation of the extracted data, and the user is not required to define a representation explicitly. On the other hand, this approach does not consider the user's preference, as the visualization relies only on the extracted data. In addition, the use of a proprietary model, such as the one provided by OpenAI, raises privacy issues, as the extracted data is transmitted in plain text to the model and the company.

In the alternative workflow, there is a single model instance that takes care of both creating the query and proposing the graphical visualization. This approach has the advantage of being faster and cheaper since only one API call for each request is performed. Another advantage is that users are free to choose their preferred data visualization format by including such information in their request. However, this workflow does not rely on the extracted data to suggest the most suitable visualization; it can rely only on the user's request and the database schema. Such an approach may result in visualizations that are not adequate to the actual extracted data.

5 Use Case

This section presents a use case to demonstrate the proposed system's applicability in supporting workers in an analysis laboratory where collaborative robots are used to handle test tubes with the help of reagents and specific manipulations to obtain desired preparations and solutions. The main objective of this use case is to illustrate how the system can cater to different types of users, based on their roles and the complexity of their requests, by offering tailored visualizations.

The laboratory staff faces diverse challenges depending on their responsibilities. On the one hand, management personnel require customized metrics on production, performance, and economic aspects, often presented in specific visualization formats. On the other hand, operational staff are focused on quality control and product monitoring, seeking information such as how many products meet quality standards, how many products have been rejected, or how often a particular reagent is used.

To address the variety of needs, users can be classified according to their data requirements and their role within the organization:

- *(a1)* **Management Level**: Users interested in tailored metrics and specific visualizations related to production, performance, or economic efficiency.
- *(a2)* **Operational Level**: Users primarily concerned with product control and quality, focusing on metrics such as compliance with standards or reagent usage.

Additionally, users' requests can be categorized based on their complexity:

- *(b1)* **Simple Data Request**: Requests for straightforward data, typically in table format or as a single value, reflecting what is directly available in the database (e.g., task lists, average task duration, or tasks completed during a day).
- *(b2)* **Complex Data Request**: Requests for more advanced metrics that involve aggregating data from various database tables and sources (e.g., products discarded due to specific reagent batches, production chain efficiency, or reagent quality based on supplier data) or that should be displayed in a specific visualization format.

The combination of user role and request complexity is shown in Table 1, which outlines the most suitable workflow for each scenario:

Table 1. User/Data Request Matrix

USER / DATA REQUEST	*(b1)* **SIMPLE**	*(b2)* **COMPLEX**
(a1) **MANAGEMENT**	*(1)* Equivalent	*(3)* ONE model instance
(a2) **OPERATIONAL**	*(2)* Equivalent	*(4)* TWO model instances

This matrix illustrates that simple requests are handled similarly across user types, while complex requests necessitate different workflows based on the user's role. The following examples illustrate the practical applications of each combination:

- **Example *(1)*:** A manager requests data on how many orders were processed in the previous month. Since this is a simple data request, both workflows would provide an equivalent result.
- **Example *(2)*:** A worker at the operational level asks for data on how many products were rejected during yesterday's quality control checks. Similar to the first case, this is a simple data request that can be handled equivalently by both workflows.
- **Example *(3)*:** A manager requests the number of completed and failed tasks for each robot over the past three months, including the average duration of both completed and failed tasks. The manager adds that the requested output should be displayed in a bar chart with one bar for each robot showing the number of completed and failed tasks, as well as additional bars for the average task duration. Since the manager knows the exact visualization format they need, the one model instance is more suitable to deliver the requested bar chart. Figure 5 shows a screenshot of the application for this example, while the SQL query generated on the user's request is reported in Listing 1.1. However, if the manager had been unsure of the best way to present the data, the two model instances could have suggested a more appropriate visualization.

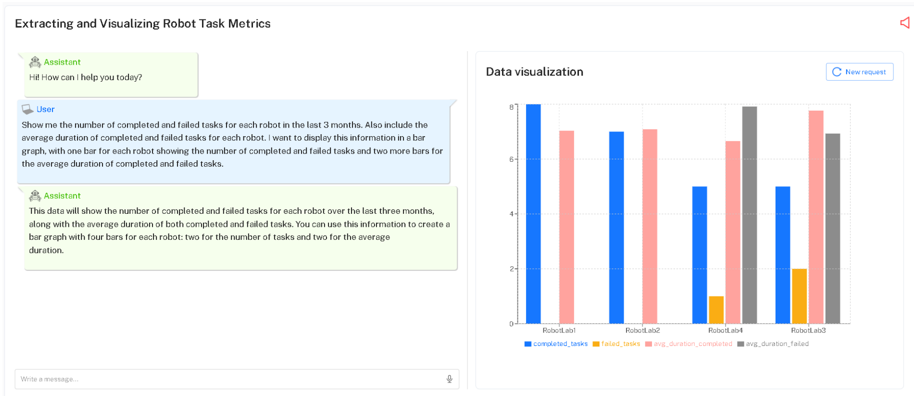

Fig. 5. A production manager requests the number of completed and failed tasks for each robot over the past three months, along with the average task durations

```sql
SELECT
  robot_name,
  COUNT(case when status = 'COMPLETED' then 1 end) as completed_tasks,
  COUNT(case when status = 'ERROR' then 1 end) as failed_tasks,
  AVG(case when status = 'COMPLETED' then duration end) as
     avg_duration_completed,
  AVG(case when status = 'ERROR' then duration end) as avg_duration_failed
FROM
  robot_tasks_execution
WHERE
  start_time between '2024-06-19' and '2024-09-19'
GROUP BY
  robot_name
ORDER BY
  completed_tasks desc;
```

Listing 1.1. Showing a bar graph with last three months production in the laboratories

- **Example *(4)*:** A laboratory technician needs to determine how frequently new pH sensors should be ordered, and requests data on the use of pH sensors in tasks performed over the past six months. In this case, the workflow with two model instances is more suitable. The first model instance initially suggests a table as the visualization format. However, the second instance, which has a more comprehensive understanding of the data, recommends a line chart as a more suitable way to represent the data over time. Figure 6 reports a screenshot of the application for this example, while the generated SQL query is reported in Listing 1.2.

```sql
SELECT manipulated_objects, start_time
FROM robot_tasks_execution
WHERE tool ILIKE '%pH sensor%'
    AND start_time >= NOW() - INTERVAL '6 MONTH';
```

Listing 1.2. Showing the use of a pH sensor as tool in the tasks performed during last 6 months

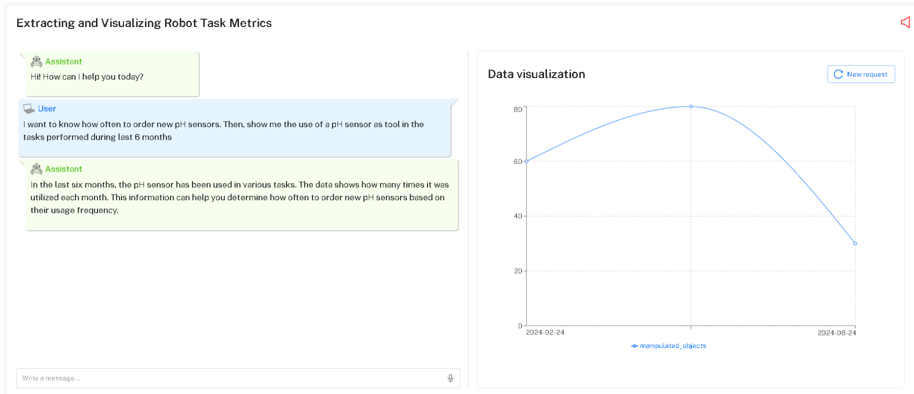

Fig. 6. A laboratory technician requests information on how often pH sensors are used, in order to decide how frequently to reorder them

This use case has shown how the proposed system can effectively support both management and operational staff in a laboratory environment by providing tailored workflows based on the complexity of their data requests. The one model instance is best suited for users with specific visualization needs, while the workflow with the two model instances excels in situations where users benefit from system-generated suggestions about data visualization.

It is important to remark that examples and results reported in this section are based on the testing we have done by executing different types of requests with different workflows, and that extensive experimentation with real users is reserved for future work.

6 Discussion and Conclusion

The prioritization of easy-to-use interfaces to access robot data paves the way for seamless human-robot collaboration, thus ensuring that the transformative potential of collaborative robots can be exploited in different work environments. In an era where knowledge is power, the capacity to comprehend and exploit robot usage data can be a pivotal factor in the success and sustainability of organizations and societies.

This paper explores how LLMs can help designers achieve the goals of HWID, using AI to improve workers' wellbeing. Specifically, it elaborates on how human workers can retrieve information from a database with the assistance of an LLM-based system, aiming to obtain more effective exploitation of collaborative robots in work contexts.

Nevertheless, only once the prototype has been tested with users will it be possible to validate the efficacy of the proposed approach and which of the two workflows is best suited to users' requests in different cases. In this sense, the examples analyzed in the use case can be a good starting point for a user study.

A further dimension of analysis for the developed prototype will be that of query optimization. This includes an examination of the proposed query efficiency in terms of execution speed. It is also crucial to assess the model's comprehension of the database structure in the context of large databases. The database structure description may influence the *context window* of the model, which determines the amount of information that the model can take into account before ignoring older messages of a conversation, thereby losing some contextual information.

An additional extension is to train the model further by performing an initial fine-tuning with different queries. Additionally, also the subsequent interactions with users, which are appropriately marked as correct, can be used to further train the model to enable gradual improvement with use. To facilitate a more comprehensive understanding of the context and the interconnection of data by the model, it will be interesting to examine the use of specific domain knowledge using the Retrieval-Augmented Generation (RAG) [15]. Using RAG, the LLM can be trained with the provision of documents and information, which will increase its ability to make connections between data and provide more accurate queries. It is important to note that this documentation should not be limited to technical information about the data itself. Rather, it must also include information about the real applications of the data. Finally, as a prospective development, it would be of interest to investigate the extension of this approach to the analysis of NoSQL databases, in which there is no predefined data structure.

In summary, this study aims to provide insights for developing tools and methodologies that enhance transparency, comprehension, and data usage by leveraging natural language interaction. The approach can be generalized and applied to other work environments beyond computing metrics for collaborative robots. This will just require changing the initial context of the LLM and its database schema, thus making the proposed solution flexible and dynamic.

References

1. Abdelnour-Nocera, J., Baricelli, B.R., Lopes, A., Campos, P., Clemmensen, T.: Human Work Interaction Design: Analysis and Interaction Design Methods for Pervasive and Smart Workplaces: 4th IFIP 13.6 Working Conference, HWID 2015, London, UK, June 25-26, 2015, Revised Selected Papers, vol. 468. Springer (2015)
2. Bi, Z.M., Luo, C., Miao, Z., Zhang, B., Zhang, W.J., Wang, L.: Safety assurance mechanisms of collaborative robotic systems in manufacturing. Robotics and Computer-Integrated Manufacturing **67**, 102022 (2021)
3. Calò, T., De Russis, L.: Leveraging large language models for end-user website generation. In: Spano, L.D., Schmidt, A., Santoro, C., Stumpf, S. (eds.) End-User Development. pp. 52–61. Springer Nature Switzerland, Cham (2023)
4. Clemmensen, T.: Human work interaction design: A platform for theory and action. Springer (2021)
5. Doyle-Kent, M.: Collaborative robotics in industry 5.0. Ph.D. thesis, Technische Universität Wien (2021)
6. Fogli, D., Gargioni, L., Guida, G., Tampalini, F.: A hybrid approach to user-oriented programming of collaborative robots. Robotics and Computer-Integrated Manufacturing **73**, 102234 (2022)
7. Galin, R.R., Meshcheryakov, R.V.: Human-robot interaction efficiency and human-robot collaboration. In: Robotics: Industry 4.0 issues & new intelligent control paradigms, pp. 55–63. Springer (2020)
8. Gargioni, L., Fogli, D.: Integrating chatgpt with blockly for end-user development of robot tasks. In: Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction. pp. 478–482 (2024)
9. Gargioni, L., Fogli, D., Baroni, P.: Designing human-robot collaboration for the preparation of personalized medicines. In: Proceedings of the 2023 ACM Conference on Information Technology for Social Good. pp. 135–140 (2023)
10. Huang, F., Li, G., Zhou, X., Canny, J.F., Li, Y.: Creating user interface mock-ups from high-level text descriptions with deep-learning models (2021)
11. Huang, S., Wang, B., Li, X., Zheng, P., Mourtzis, D., Wang, L.: Industry 5.0 and society 5.0—comparison, complementation and co-evolution. Journal of manufacturing systems **64**, 424–428 (2022)
12. Jiang, E., Toh, E., Molina, A., Donsbach, A., Cai, C.J., Terry, M.: Genline and genform: Two tools for interacting with generative language models in a code editor. In: Adjunct Proceedings of the 34th Annual ACM Symposium on User Interface Software and Technology. p. 145–147. UIST '21 Adjunct, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3474349.3480209
13. Kanazawa, A., Kinugawa, J., Kosuge, K.: Adaptive motion planning for a collaborative robot based on prediction uncertainty to enhance human safety and work efficiency. IEEE Transactions on Robotics **35**(4), 817–832 (2019)
14. Kolvig-Raun, E.S., Kjærgaard, M.B., Brorsen, R.: Joint stress estimation and remaining useful life prediction for collaborative robots to support predictive maintenance. IEEE Robotics and Automation Letters (2024)
15. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems **33**, 9459–9474 (2020)
16. Li, J., Hui, B., Qu, G., Li, B., Yang, J., Li, B., Wang, B., Qin, B., Cao, R., Geng, R., et al.: Can llm already serve as a database interface. A big bench for large-scale database grounded text-to-sqls. CoRR abs/2305.03111 (2023)

17. Mahmud, T., Hasan, K.A., Ahmed, M., Chak, T.H.C.: A rule based approach for nlp based query processing. In: 2015 2nd international conference on electrical information and communication technologies (EICT). pp. 78–82. IEEE (2015)
18. Proia, S., Carli, R., Cavone, G., Dotoli, M.: A literature review on control techniques for collaborative robotics in industrial applications. In: 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE). pp. 591–596. IEEE (2021)
19. Rahman, M.M., Khatun, F., Jahan, I., Devnath, R., Bhuiyan, M.A.A.: Cobotics: The evolving roles and prospects of next-generation collaborative robots in industry 5.0. Journal of Robotics **2024**(1), 2918089 (2024)
20. Rajkumar, N., Li, R., Bahdanau, D.: Evaluating the text-to-sql capabilities of large language models. arXiv preprint arXiv:2204.00498 (2022)
21. Rozo, L., Calinon, S., Caldwell, D.G., Jimenez, P., Torras, C.: Learning physical collaborative robot behaviors from human demonstrations. IEEE Transactions on Robotics **32**(3), 513–527 (2016)
22. Sahoo, S., Lo, C.Y.: Smart manufacturing powered by recent technological advancements: A review. Journal of Manufacturing Systems **64**, 236–250 (2022)
23. Vemprala, S., Bonatti, R., Bucker, A., Kapoor, A.: Chatgpt for robotics: Design principles and model abilities. Tech. rep., Microsoft (2023), https://www.microsoft.com/en-us/research/uploads/prod/2023/02/ChatGPT_Robotics.pdf
24. Wannasuphoprasit, W., Gillespie, R.B., Colgate, J.E., Peshkin, M.A.: Cobot control. In: Proceedings of International Conference on Robotics and Automation. vol. 4, pp. 3571–3576. IEEE (1997)
25. Zhang, X., Khedri, K., Rawassizadeh, R.: Can llms substitute sql? comparing resource utilization of querying llms versus traditional relational databases. arXiv preprint arXiv:2404.08727 (2024)
26. Zizic, M.C., Mladineo, M., Gjeldum, N., Celent, L.: From industry 4.0 towards industry 5.0: A review and analysis of paradigm shift for the people, organization and technology. Energies **15**(14), 5221 (2022)