Runtime Framework for Smartification Projects

Pedro AMARAL
DEE, NOVA School of Science and Technology,
NOVA University of Lisbon, Caparica 2829-516, Portugal

Jorge S. CALADO
UNINOVA-CTS, NOVA School of Science and Technology,
NOVA University of Lisbon, Caparica 2829-516, Portugal

João SARRAIPA
UNINOVA-CTS, NOVA School of Science and Technology,
NOVA University of Lisbon, Caparica 2829-516, Portugal

Ricardo JARDIM-GONÇALVES
UNINOVA-CTS, NOVA School of Science and Technology,
NOVA University of Lisbon, Caparica 2829-516, Portugal

ABSTRACT

Industry 4.0. is introducing the process of Smartification, which is the process of adding technology to everyday, non-technological objects, enhancing them, and granting them new uses. And since said smartified objects can be accessed, and controlled remotely through computers, it opens up the chance to enhance equipments during runtime. Best assets provided during runtime include monitoring the correct execution of the program and signaling the maintenance system when needed. This system can also be connected to a platform capable of giving the user enough information about the equipment, and tools to report back feedback to developers. This data, along with normal user records is stored in a database, and further data studies and analysis can be performed to profile users and suggest further upgrades through updates. The goal is to present a framework capable of providing the main considerations, concerns and concepts for runtime exploration capable of enhancing and adapting any equipment or object.

Keywords: Runtime, Smartification, Data, IoT, Framework, Data Model, Monitoring

1. INTRODUCTION

The Smartification process consists in enhancing and adding technology to already existing, non-technological objects. With this we are in fact providing objects with "smart capacities" same as we did with phones and watches previously[1]. A remark on this reality is the capacity to control different devices and objects independently of its domain, directly by accessing them from computers and smartphones that are connected to the web. With the increase in popularity of technologies such as industry cloud, artificial intelligence, internet of behaviors, privacy-enhancing computation[2], [3], that were popularized, facilitated and made cheaper by Industry 4.0.[4], smartification solutions are becoming more popular and conquering more industries, including furniture design and manufacturing industry, namely the use case being explored in this paper.

Motivation
Although the above-mentioned smartification growth is mostly beneficial, this fast-paced smartification of reality can also be the root for some societal problems. Companies are discovering more and more ways to turn data into valuable knowledge about their clients and needs [5].

Gartner's survey conducted in 2020, regarding IoT projects, showed 35 percent of respondents were selling or planning to sell data collected by their products and services[2], which represents the value of exploring devices during runtime to gather meaningful data, as well as enforcing protocol security and ethics, while keeping them up-to-date. According to the survey, organizations are lacking in these departments, by having unclear legislations and policies, and insufficient technology to ensure privacy rights are upheld, this means that stored data can become a liability for companies if security protocols are not followed or are not up to date.

Moreover, devices present themselves as a liability when lacking seamless real-time connectivity with its environment, which creates small technological islands of data. Either because they are incapable of communicating with their surrounding devices due to incompatibility issues, or because big tech manufacturers are eager to lock-in users by making their devices only communicate among peer devices designed by the same company [6].

Hence the need for a framework common to all the devices that upholds these regulations, which leads to companies and developers to not be as concerned with security and compatibility issues, that brings more focus to other advantages achieved through smartification, such as, monitoring the device's execution, evaluate the performance and the capacity to connect the device to the internet, which enables real-time hotfixes and the introduction of new features with a continuous integration and continuous delivery (CI/CD) approach.

It is the authors belief, that a common agreed upon framework can also catalyze the growth of smartification. Said smartification can even be expanded to the scope of smart cities, where it is possible to automate shipping according to opening hours, or with the creation of digital information hotspots, accessed by QR codes, and coordinating the pedestrian paths according to traffic lights timings, when there are multiple options to cross the road. Through multiple sensors in contact with the skin, such as a smart watch, or vibration sensors at home to detect falls from their residents, This sort of smartification can also save human lives, when used to monitor vital signs in people[7][8]. Such advancement being a clear representation of

ISBN: 978-1-950492-61-9
ISSN: 2771-5914

https://doi.org/10.54808/IMCIC2022.02.74

an increase in quality of life to habitants of multiple cities across the globe.

Contribution
To tackle these concerns and consider the new possibilities, this paper suggests splitting runtime exploration into four broader topics, namely monitoring, maintenance, profiling and updating. It also explores a framework with support from the Inedit Project[9] in order to facilitate similar endeavors in the future. The project this paper is inserted into, Inedit, also provided conditions to thoroughly test the implementation in real life application of a smart cabinet. This device task is to open the cabinet with voice command or when someone is nearby, to facilitate the mundane activity of storing a big enough object that requires both hands.

The mentioned tests include validating the Appendix 1- Data Model by simulating hypothetical data curated in lab that aided in the model creation, in which it is now possible to store all the predicted information and making the model easily scalable when in need to incorporate a new scenario. Namely to store new updates data regarding maintenance and new features added, listing the sensors used and their values during monitoring, and to gather user information privately, by detaching personal information from collected data. It is also tested that a notification is sent to the admin if a sensor reads a value that is parameterized as faulty behavior from the device. However, a limitation found during the project was the profiling of users, since the data is simulated and not derived from real continuous use.

The goal is to understand if, and by how much can a furniture piece be enhanced with runtime exploration. And if verified, to compile a framework with the main considerations, concerns and concepts for runtime exploration in a furniture piece, but remaining generic enough that they could also be applied to the other industries.

2. STATE OF THE ART

This section describes what technology is available in the scope of smartification and which approaches have been considered. Namely the approaches made in monitoring, data or user profiling, and security respectively.

Monitoring a device consists of several "verification techniques that allow checking whether an execution of a system under scrutiny satisfies or violates a given correctness property" [10]. Which suits the purpose of creating better, and overall, more robust systems capable of adaptation and self-repair. If that is not enough, trigger functions as a response to an event are more than capable of automating the monitoring process and exchange of data.

Data is a growing industry[2] that is becoming more and more expensive due to its increase in value to different enterprises. Companies are realizing that besides knowing their target audience, it is also possible to organize their users and clients, by associating them into categories. With users already categorized, it is then possible to redirect advertising only to specific types of users. On the one hand buying the user profiles is not cheap, however companies save money by marketing products to less, but more interested clients. Gartner predicts that by 2022, 35% of large organizations will be either selling or buying data from online data marketplaces, up from 25% in 2020 [11]. Therefore, if data is becoming essential to business, a secure environment to transfer and store data, must be ensured, and up to date. According to studies conducted by Gartner in 2019, users trust in companies has decreased while privacy invasion increased[12].

Security and encryption of data has become a priority for companies that want to thrive in the market, so the General Data Protection Regulation (GDPR) entity is in charge of creating and enforcing laws of data protection[13]. This way data can be gathered safely and be a very important part of technology enterprises and increase users' quality of service.

Data gathered and shared will form the basis for Knowledge Discovery, the process of extracting knowledge from data. It is of uttermost importance to adequately use information gathered to further enhance the already established device using updates, or to assess the user experience during runtime. This allows developers to create a better, and up to date version of a product. This process of identifying patterns and organizing users into different categories according to their specific set of characteristics, called Data Profiling, helps erasing two problems of Information Technology: (1) the overwhelming quantity of information that more and more devices are creating and having to store; and (2) the lack of a clear distinction between noise, information and knowledge[14]. Because data without withdrawing knowledge or a defined conclusion from it is obsolete data that wastes digital resources.

It is possible to conclude then, that a framework compiling solutions to the problems above mentioned of facilitating data gathering, storing, following data security regulation regarding smartified objects security, monitoring and user profiling is beneficiary, while including some other new aspects, such as maintenance and gathering user feedback for the equipment. This paper then suggests guidelines and a data model capable of not only effectively storing the information attached with said framework but also to quickly formulate a prototype.

3. FRAMEWORK AND DATA MODEL

This section identifies and details the main topics of exploration during runtime, which briefly present the concepts and portray the specification of each major categories. A rundown of the interactions among all these categories is also clarified in the end of the section with a diagram.

According to the research done, most of the exploration during runtime can be categorized into four main topics. Namely monitoring the correct execution of the program, signaling the correct maintenance once equipment's behavior is not according to norms. This, while having a platform capable of giving the user enough information about the equipment and tools to report back feedback to developers. Moreover, this data, along with normal user records will be saved in a database so that further data studies and analysis can be performed in order to profile users and suggest further upgrades and updates accordingly.

A data model was developed taking in mind the presented requirements for runtime. However, in order to fit numerous different projects and small differences among them, while still attending to runtime specifications, the data model had to remain abstract enough. The latter, however, does not prohibit further refinements for specific projects. It is encouraged to use the Data Model as a foundation for an individual Data Model specifically designed for a given project. The template is provided in Appendix 1- Data Model.

Monitoring

Taking a deeper look at each topic, Monitoring will be highly dependent on the equipment that is going to be monitored. This happens since monitoring is basically establishing rules and thresholds for sensor reads besides naming and checking for behaviors that are categorized as expected or non-conflictual.

To achieve that, the Data Model suggested will use the components listed as *Measurements* to receive and store input from the *sensors* and associate it with an *activity*, or in other words the status or function the equipment is performing during the sensor measurement. The list of sensors and activities possible for the project should be listed in *sensor* and *activity* respectively in order to correctly link it with a *measurement.*

An easy to understand and practical application would be monitoring the battery that should be replaced before it gets depleted, or any sensor that starts outputting values that are far from the expected ones.

Maintenance

From the last sub section, we can then understand how dependable *maintenance* is to *monitoring*. Using, as an example, the monitoring of a battery, the maintenance log can be automated. Once the battery level falls below 10%, the user is notified until the battery is replaced.

On the other side of the spectrum for Maintenance is *manual_maintenance*, for exceptional cases where the equipment has a malfunction or stops working completely. In this case a user interface is employed to facilitate reporting the situation to be fixed, and once it is in the system someone in charge for the maintenance of the project will receive a notification and fix the problem as soon as possible. Which in the Data Model works the same as *automatic Monitoring / Maintenance,* except the input is generated manually through the *user feedback*, even though the flagging for maintenance is still achieved by automatic trigger events.

Another case for Maintenance is when users manually report a reoccurring *bug,* where the equipment might require a hotfix achieved with an update patch. Failing to correct the problem significantly increases users' dissatisfaction with the product and might even create an obsolete device.

In the Data Model, two distinct components of update are present. *Update project* will keep track of any big change to the project or equipment in itself, whereas *update components* is meant to keep track of updates to single components belonging to the device.

According to the project, the developing team can choose to either use the data model to only keep track of the most recent version, with the patch notes written on the description, or to incorporate the file into the database. In the first option however, there must be some information regarding the whereabouts of the specific file, in the Data Type that is more convenient.

User Feedback

User Feedback stands as an invaluable source of information to the project. Developers must consider the users' feedback to seek improvement and make sure all the users' needs are met. Since there is never too much information about the user experience, forms should be very thoughtful and well prepared to avoid excessive questioning but extracting as much information as possible, while encouraging the user to take its time to do simple feedbacks trough online simple forms, or written reviews.

Correctly processing this information, means that the Data Model categorizes the components according to the submitted information. Which also facilitates if there is a need to expand the data model when there is more information to be acquired through users' experience and feedback Presented in the data model and considered invaluable are users suggested new *features, rating* users' satisfaction with the product and *bug* & *malfunction* reports.

Profiling

Lastly but not less important, the profiling of its users also proves as a major advantage that runtime exploration offers. Using the amount of data that was gathered, Knowledge Discovery algorithms should be implemented in order to recognize patterns among different users and correctly categorize them. This ensures that developers are always in touch with the users' needs and can better develop updates accordingly. Another advantage of profiling users is associating certain behaviors to different users, furthermore associating malfunctions to these same behaviors. With this information developers can more accurately predict what kind of problems specific users might have in the future and prevent them with timely maintenance.

The Data Model components for *profiling* include both *behaviors* and *characteristics*. The list of identifiable traits will be inserted into *characteristics,* whereas *behaviors* is meant to describe different actions one equipment can perform according to its profile, or in other words, the user.

The following diagram, portrayed in figure 1 serves the purpose of illustrating the main responsibilities of each topic and how they interact with each other.

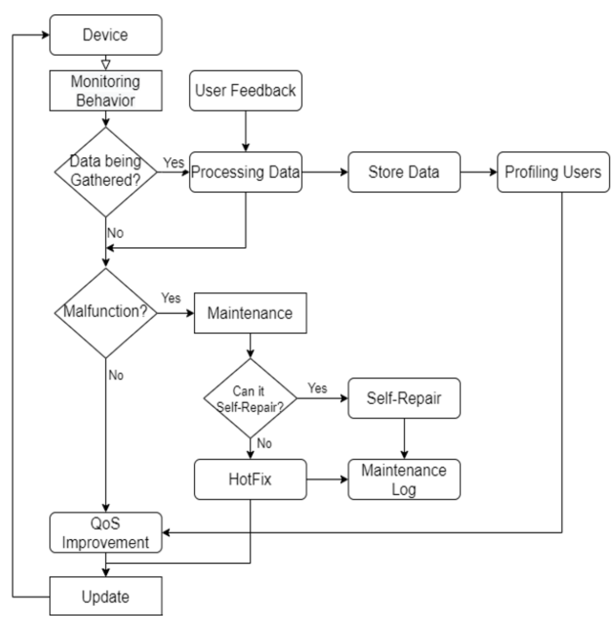


Figure 1 – Framework Flowchart

After providing the device with every singular category of runtime previously listed and detailed, developers must make sure all the interactions among the different runtime categories are ensured, by revisiting the above diagram. When following the flowchart, any device should be able to monitor its own execution and functions, gather and process all the relevant information during runtime, fix any malfunctions that were identified, either digitally with a software update, or physically by correctly notifying the developers/manufacturers appointed to mend it. And lastly, if possible, to update the device, searching for a quality of service improvement for the end user.

4. DEMONSTRATION

After elaborating this framework, it was tested in real-time, as a follow-up of work already developed by a Prototyping System, which is another entity that collaborated in this project in the scope of smartification, specifically responsible to design a prototype capable of triggering the opening of a door using voice commands and/or when a user is approaching the cabinet door. Only with said device was it possible to conduct the current implementation during runtime as a proof of concept for this framework. This implementation was developed using a proximity sensor, an Arduino Yun, an actuator, and its respective control board.

The main goal for this equipment during runtime was to monitor the correct execution of the prototype, and correctly identify what data should be stored and allow this data to be presented and interacted with, by users' or developers through a graphical interface. If so, the implementation suggested by this framework would be providing useful information to both the manufacturers, developers and end users. Hence, refining the equipment with runtime exploration, as proposed in sub-section "Contribution".

Monitoring
To correctly monitor this project, the Arduino sends messages containing the values read from sensors and other relevant information, in *JSON* format, through an *MQTT* protocol to a *server* developed in *Node.js,* the latter stores this data in the database. which is displayed in a user graphical interface. If any of the values gathered do not fall under pre-established, expected threshold values, (in this case defined with *SQL trigger functions),* the equipment is flagged for maintenance.

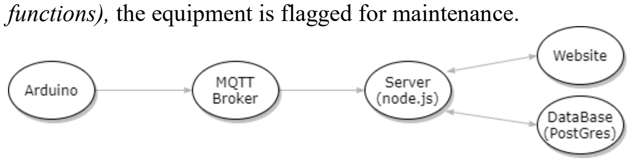


Figure 2 – Demonstration architecture

Maintenance
When the device has correctly identified a problem with itself, the user is automatically notified once he accesses the webpage, and is redirected to the maintenance page portrayed in figure 3. This page is programmed to only display the sensors in need of maintenance: in this case, the battery. And after submitting the information, the server will change the information stored in the database accordingly.
Other options could be contemplated for the future, such as sending an e-mail to notify the user, if the priority or maintenance requirement justifies it. Additionally, if the server has not been contacted in some time, the system will automatically acknowledge that something is not working correctly and notify the user.

There is also the option to perform over-the-air updates since the Arduino being used is Arduino Yun, with access to the internet. These updates consider bugs and malfunctions reported through user feedback, and solve them through hotfixes to the software, that will be uploaded into the board.

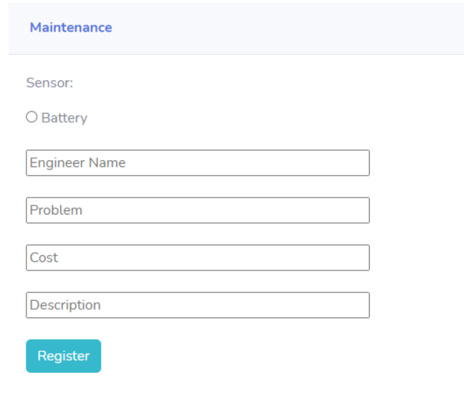


Figure 3 - Maintenance interface

User Feedback
The relevant data is also displayed in a graphical interface on the web seen on figure 4. In the top left container, the website tracks the state of the Equipment and in the bottom left it displays the list of sensors, with the most recent measure and a color-coded risk assessment. The list of sensors are the battery, proximity that evaluates the distance from the user till the door. The top right container has a link to an online survey to submit feedback on the equipment.

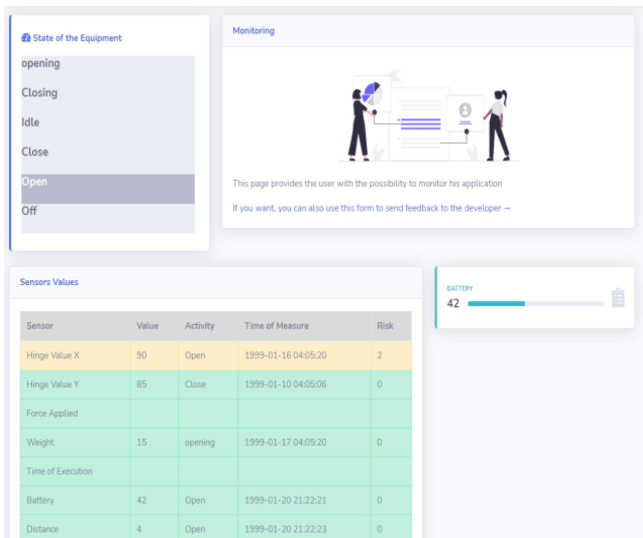


Figure 4 -User Interface

The user feedback forms were implemented in Google Forms[15] and contain generic questions regarding the device performance and user satisfaction with the equipment, such as for how long has the device been used, any malfunction report, or suggestion for a new feature.

Profiling
Future work includes clustering the data collected from equipment's installed and used by different people into profiles. This, however, requires a large enough Data set and Knowledge Discovery algorithms to be applied to said data.

In this particular case, information regarding the amount, frequency, and speed in which the door is opening (and closing) is being tracked. This data can later on be correlated with malfunction reports to predict and prevent the same damage in peer devices.

Also, if multiple users decide to use the same device, the equipment can also enable profiling each user's settings and/or behaviors, according to different needs. In this example, one user could favor a bigger time delay between opening and the closing of the cabinet, and the Arduino is capable of fetching the information from the database.

5. CONCLUSIONS

The proposed framework for runtime exploration proved valuable when tested in the exemplified use case, showcasing the ability to quickly fabricate a prototype and upgrade an existing furniture piece into a smart object capable of integrating and interacting with a smart ecosystem in addition to receiving information from it. The goal of this framework was then accomplished, since it increased the speed in which a new smartification project was developed and presented, providing useful input in what should be considered, while also providing a template for the core functions of an equipment. It was possible to correctly automate monitoring, signal the need of maintenance and add new features, safely and in a real time manner. All of these processes were also stored in a proposed generic Data model. Said data is then used to further understand malfunction reports and quite possibly prevent them in other devices.

Future work will include further optimization to the use of data gathered through machine learning and knowledge discovery algorithms to profile users, making the exploration of smartified objects an asset in technology development for smart ecosystems. The research and framework can also be verified within the industry with different smart objects with increasingly complexity levels. Such as a smart door that provides bigger challenges in security, or upgrading already existing designs such as a chair, a table[8] or include different features in a cabinet, such as tracking of its components. Such upgrade could also be used to branch out into more challenging and complex projects such as smart fridge or a smart pantry.

The framework and Data Model were deployed in a *Docker* container and uploaded to online repositories[16], to be easily available and ready to be worked on with all the correct dependencies and libraries.

6. ACKNOLEDGEMENTS

The authors acknowledge the European Commission for its support and funding, and the partners of the research project, INEDIT, DT-FOF-05-2019, under the grant agreement number 869952, and also Plurianual – UIDB/00066/2020.

7. REFERENCES

[1] G. Schuh, V. Zeller, J. Hicking, and A. Bernardy, "Introducing a methodology for smartification of products in manufacturing industry," *Procedia CIRP*, vol. 81, pp. 228–233, 2019, doi: 10.1016/j.procir.2019.03.040.

[2] G. Omale, "Gartner Identifies Top 10 Strategic IoT Technologies and Trends," 2018.

[3] K. Costello and M. Rimol, "Gartner Identifies the Top Strategic Technology Trends for 2021," 2020.

[4] A. Prosser, "What the Smart City in the Danube Region Can Learn From Industry 4.0," *Cent. East. Eur. eDem eGov Days*, vol. 331, pp. 191–201, Jul. 2018, doi: 10.24989/OCG.V331.16.

[5] O. Maimon and L. Rokach, "Data Mining and Knowledge Discoveru Handbook," vol. 40, no. 6, p. 9823, 2001.

[6] O. Bello and S. Zeadally, "Toward efficient smartification of the Internet of Things (IoT) services," *Futur. Gener. Comput. Syst.*, vol. 92, pp. 663–673, 2019, doi: 10.1016/j.future.2017.09.083.

[7] T. Erlina, M. R. Saputra, and R. E. Putri, "A Smart Health System: Monitoring Comatose Patient's Physiological Conditions Remotely," *2018 Int. Conf. Inf. Technol. Syst. Innov. ICITSI 2018 - Proc.*, pp. 465–469, 2018, doi: 10.1109/ICITSI.2018.8696094.

[8] F. Luis-Ferreira, J. Sarraipa, and R. Goncalves, "Smartification of Home Appliances for Safety Assessment and Risk Alert," *IFAC-PapersOnLine*, vol. 52, no. 10, pp. 207–211, Jan. 2019, doi: 10.1016/J.IFACOL.2019.10.065.

[9] European Commission, "Funding & tenders," *Funding & Tender Opportunities*, 2017. https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/topic-details/sc1-bhc-20a-2020;freeTextSearchKeyword=;typeCodes=1;statusCodes=31094501,31094502,31094503;programCode=H2020;programDivisionCode=null;focusAreaCode=null;cros (accessed Oct. 08, 2021).

[10] M. Leucker and C. Schallhart, "A brief account of runtime verification," *J. Log. Algebr. Program.*, vol. 78, no. 5, pp. 293–303, 2009, doi: 10.1016/j.jlap.2008.08.004.

[11] L. Goasduff, "Gartner Top 10 Trends in Data and Analytics for 2020," *Gart. Inc.*, pp. 1–5, 2020, [Online]. Available: https://www.gartner.com/smarterwithgartner/gartner-top-10-trends-in-data-and-analytics-for-2020/.

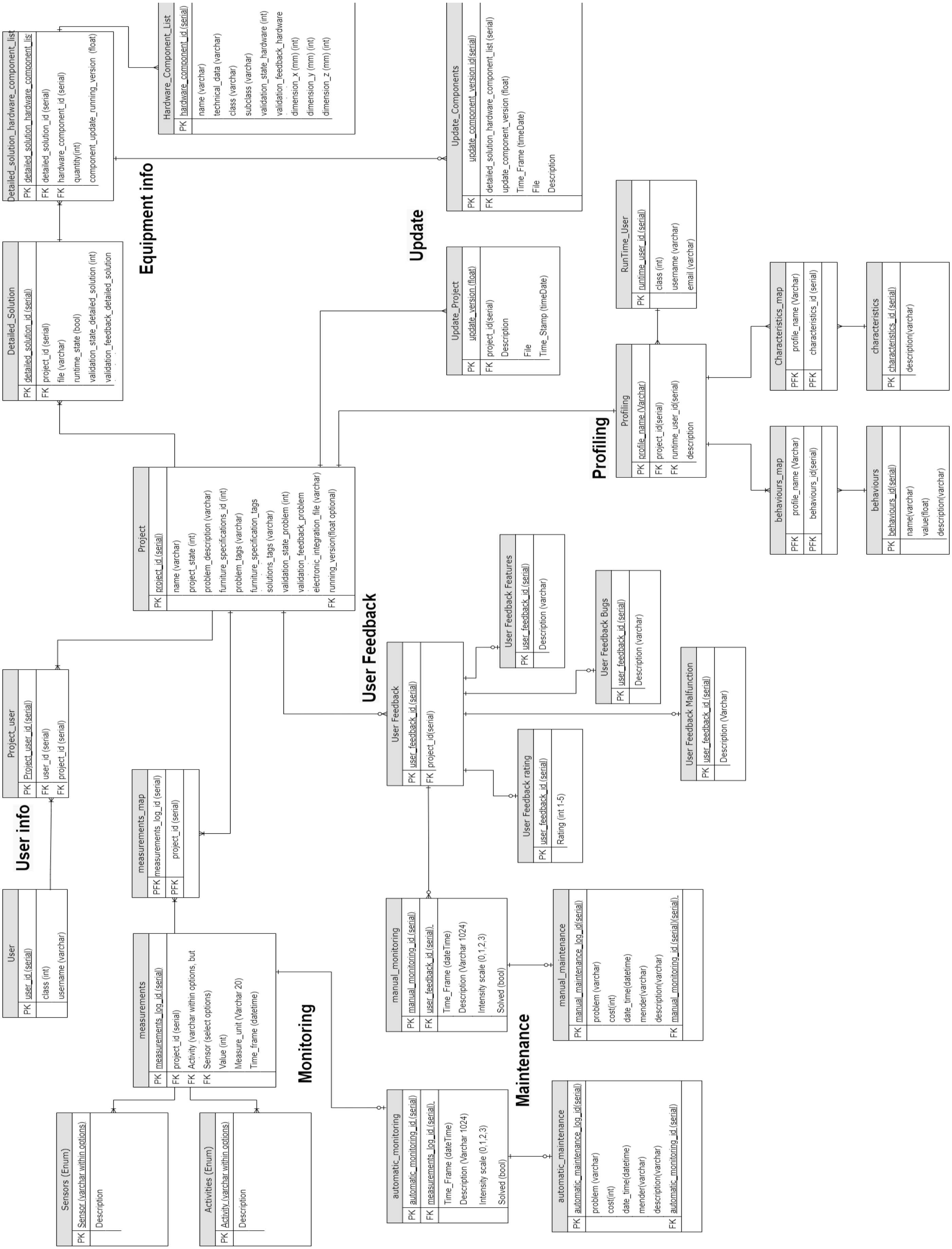
[12] S. Moore, "Gartner Predicts for the Future of Privacy 2020," *Gartner*, 2020, [Online]. Available: https://www.gartner.com/smarterwithgartner/gartner-predicts-for-the-future-of-privacy-2020/.

[13] M. Goddard, "Viewpoint: The EU general data protection regulation (GDPR): European regulation that has a global impact," *Int. J. Mark. Res.*, vol. 59, no. 6, pp. 703–706, 2017, doi: 10.2501/IJMR-2017-050.

[14] M. Hildebrandt, "Profiling: From data to knowledge: The challenges of a crucial technology," *Profiling From data to Knowl. challenges a crucial Technol.*, vol. 30, no. 6, pp. 548–552, 2006, doi: 10.1007/s11623-006-0140-3.

[15] P. Amaral, "Runtime User Feedback form." https://docs.google.com/forms/d/e/1FAIpQLSfm6HfKkzYcOiMGZXGJO0CfZ7mUJclj--3jBRu1cSAI8TmKRQ/viewform?usp=sf_link.

[16] Amaralesco, "Smartification Repository." https://github.com/Amaralesco.



Appendix 1- Data Model