Discussion Topic: OO Design for IoT

1. **Referring to the article by Fortino et al. (2015), consider the strengths and weaknesses of designing a metamodel to support the object-oriented design of the IoT. Design a smart model equivalent to that presented in Figure 6, which would instead support the operation of a driverless car.**

In this discussion, I will be examining the advantages and disadvantages of using a metamodel to facilitate the design of object-oriented IoT systems, as outlined in the article "Towards a Development Methodology for Smart Object-Oriented IoT Systems: A Metamodel Approach" by Fortino et al. (2015). Additionally, at the end of this discussion, I will present a smart model similar to the one shown in Figure 6 in the article to support the operation of a driverless car.

Fortino et al. (2015) suggest that using metamodels for analysis, design, and implementation is a modern approach to constructing Smart Object (SO) based systems. It is worth noting that the software engineering methods utilised for developing IoT systems are still in their initial stages. To support they provide a case study of a smart office SO.

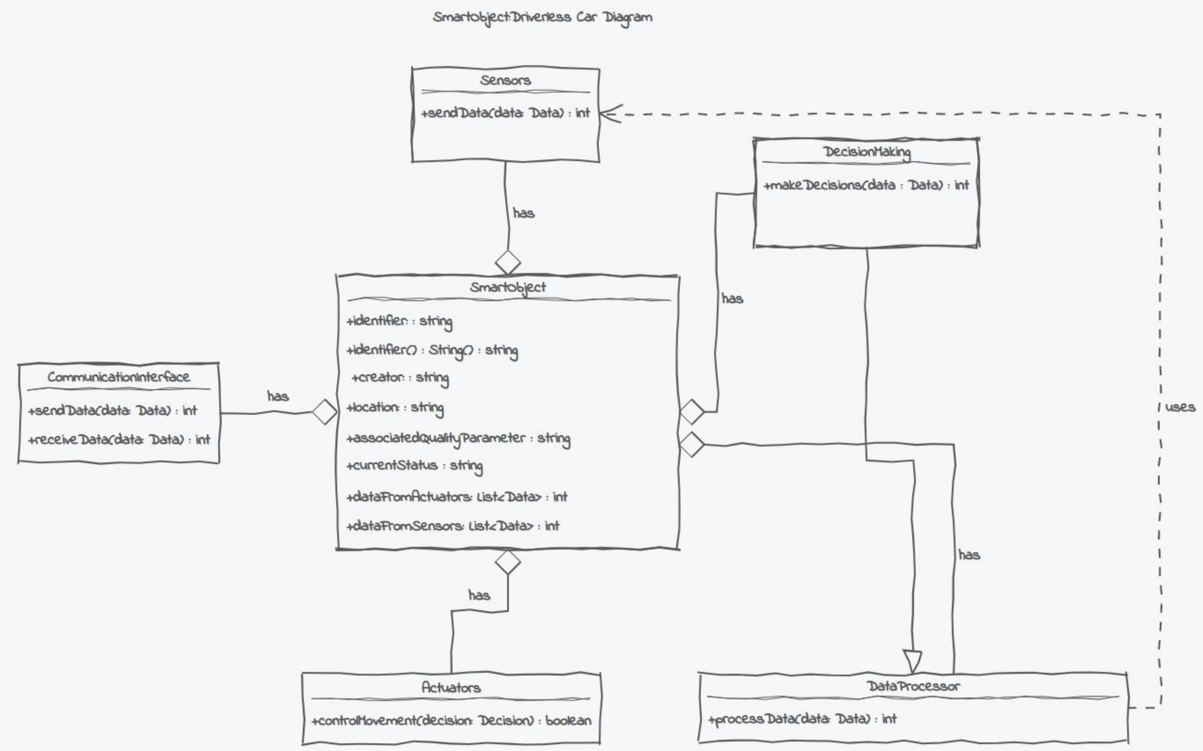
IoT metamodels are essential for IoT systems' easy reuse and interoperability (Amjad et al., 2022). Employing reusable components becomes effortless when using standardised and transparent frameworks, as emphasised by Fortino et al. (2015). Fortino et al. (2017) found that this method promotes communication and compatibility between IoT products and services.

Metamodels are advantageous for IoT systems that evolve with new devices and technologies. As Kashmar et al. (2022) stated, they are highly adaptable, allowing quick modifications without disrupting the entire system. This facilitates easier integration of new functionalities and improves scalability (Leminen et al. 2012) as the IoT ecosystem develops.

Creating a metamodel has advantages, but there are also some downsides. It is not easy to develop a comprehensive and accurate metamodel that considers all conceivable IoT scenarios (Udoh & Kotonya, 2018). It can be challenging to create a metamodel for many IoT applications. According to Deursen et al. (2007), maintaining a balance between generality and specificity is critical.

Metamodels for IoT applications must be designed and implemented with skill, consideration, refinement, and adherence to cutting-edge standards and technology. When developing software and making defensible decisions, it is crucial to consider these variables.

Below is my attempt to project the required metamodel to support the operation of a driverless car:



Smart objects are physical or virtual objects that use embedded processors, sensors, and connectivity to exchange data with the environment, other objects, and systems. They can enhance interactions and functionality in the physical and virtual worlds and be analysed for decision-making, efficiency, and product performance. This concept has several origins and uses and overlaps with other terms such as Smart Devices, Tangible Objects, Tangible User Interfaces, and Things in the Internet of Things (Academic Accelerator, N.D.).

And below is the explanation of the interactions between the classes in a metamodel to support the operation of driverless cars:

- The smart object is the system that controls the driverless car. It combines sensors, data processors, decision-makers, actuators, and communication interfaces.
- The car's sensors gather information about its surroundings and can operate independently or aggregate with the smart object.
- The data processor is a component of the smart object that processes sensor data and generates useful information.
- The decision maker uses processed data to control the vehicle, inheriting from the data processor and providing additional methods for handling the car.
- The actuators direct the car's movement based on the decision-maker's choices. The smart object can aggregate them, or they can exist independently.
- The communication interface connects with other vehicles and the environment, using infrastructure without ownership.

The interaction and coordination between these elements would be made easier by the software system used to build the smart model. As stated by Sarker (2021), a software system would gather sensor data, analyse it to extract useful information, use decision-making algorithms to choose the best course of action, and then regulate the actuators by that conclusion. The communication interface would also allow connectivity with infrastructure and other vehicles for real-time coordinated operations and information sharing.

References:

Fortino, G., Guerrieri, A., Russo, W. & Savaglio, C. (2015). Towards a Development Methodology for Smart Object-Oriented IoT Systems: A Metamodel Approach. *2015 IEEE International Conference on Systems, Man, and Cybernetics*. doi:https://doi.org/10.1109/smc.2015.231.

Amjad, A., Azam, F. & Anwar, M.W. (2022). *Device Interoperability for Industrial IoT using Model-Driven Architecture*. [online] IEEE Xplore. doi:https://doi.org/10.1109/INMIC56986.2022.9972976.

Fortino, G., Savaglio, C., Palau, C.E., de Puga, J.S., Ganzha, M., Paprzycki, M., Montesinos, M., Liotta, A. & Llop, M. (2017). Towards Multi-layer Interoperability of Heterogeneous IoT Platforms: The INTER-IoT Approach. *Internet of Things*, [online] pp.199–232. doi:https://doi.org/10.1007/978-3-319-61300-0_10.

Kashmar, N., Adda, M. and Ibrahim, H. (2022). HEAD Access Control Metamodel: Distinct Design, Advanced Features, and New Opportunities. *Journal of Cybersecurity and Privacy*, 2(1), pp.42–64. doi:https://doi.org/10.3390/jcp2010004.

Leminen, S., Westerlund, M., Rajahonka, M. and Siuruainen, R. (2012). Towards IOT Ecosystems and Business Models. *Internet of Things, Smart Spaces, and Next Generation Networking*, pp.15–26. doi:https://doi.org/10.1007/978-3-642-32686-8_2.

Udoh, I.S. & Kotonya, G. (2018). Developing IoT applications: challenges and frameworks. *IET Cyber-Physical Systems: Theory & Applications*, 3(2), pp.65–72. doi:https://doi.org/10.1049/iet-cps.2017.0068.

Deursen, A. V., Visser, E. & Warmer, J. (2007). *Model-Driven Software Evolution: A Research Agenda*. [online] Available at:

https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=cee95cb0c83ef06fca3bdb0ffd000b9858f00b7e [Accessed 1 Jul. 2023].

Academic Accelerator (N.D.). *Smart Object Encyclopedia, Science News & Research Review.* Available at: https://academic-accelerator.com/encyclopedia/smart-object#!

Sarker, I.H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, [online] 2(3), pp.1–21. doi:https://doi.org/10.1007/s42979-021-00592-x.