Discussion Topic: What are the strengths and weaknesses of designing a metamodel to support object-oriented design of the IoT?

My Initial Post

To me, the biggest strength of a metamodel in the context of the internet of things, is the handle it provides for navigating the domain. People only have a limited amount of time, so having only one succinct, yet exhaustive structure to learn in order to make relevant software, instead of numerous emergent models developed individually, has a benefit. Metamodeling supports component-based modelling and reusability, so I think it is an overall time save**. Do you agree that metamodeling saves time?**

As to the weaknesses of metamodeling, the one I'd like to point out is that metamodels may become outdated eventually as new technology comes into play, and eventually need to be replaced. Going by history, this is especially likely in domains such as the IoT which isn't fully established yet.

My Initial Post / Peer A Response

I agree with your point that metamodeling can save time, as it seems relatively easy to construct models compared to other modelling languages. As you have rightly pointed out, metamodeling supports component-based modelling, which promotes reusability, potentially increasing efficiency and saving time.

I also agree with you that, as with many other concepts and developments in the tech world, metamodels might eventually become outdated once new technologies emerge or the metamodel idea itself changes. For now, I believe it remains a valuable tool.

My Initial Post / Peer B Response

Hi,

I would say that metamodeling can save time if there is a standardized approach to modelling and reusing components, which can simplify the design process and make it easier to learn and apply the relevant software.

One question I have is, **what are some strategies that can be used to mitigate the risk of metamodels becoming outdated and ensure they remain useful in the long term?**

My Initial Post / Peer B Response / My Response

Very good question thank you,

One way to mitigate the risk of a metamodel becoming outdated, would be to ensure that it meets the standards of relevant international organisations.

Organisations such as 'International Organization for Standardization' (ISO), 'International Electrotechnical Commission' (IEC) and 'International Electrical and Electronics Engineers' (IEEE) define standards relevant to metamodels. For example; ISO/IEC/IEEE 42020:2019 claims to "establish a set of process descriptions for the governance and management of a collection of architectures and the architecting of entities". Certain organisations such as ISO, IEC and ISO are highly credible. The standards of ISO/IEC/IEEE particularly, are built from previous standards, which are equally technical. For example; ISO/IEC/IEEE 15288:2015, define what's meant by processes in ISO/IEC/IEEE 42020:2019. It defines an architectural process as a process that belongs to a set that "can construct system life cycle models".

From this we can conclude that **risk may be mitigated if the metamodel designer (or team) is versed in technical standards**. Purely because the designers in that case would be taking into account a large amount of research that has already been conducted on metamodeling. More generally, risk may be mitigated through peer effort.

I also think it's worth pointing out, that according to NAFv4, the NATO architecture framework, a metamodel should present:

- entities,
- attributes,
- relationships
- constraints

We can't be sure about the requirements of IoT in ten or twenty years time, but certain types of these things are fundamental to language, so are unlikely to be useless in the future. For example a 'has-a' relationship is still going to be useful in ten years time. By being general, the metamodel is more likely to be usable in the future. But to be useful, the metamodel would need higher-level concepts too. For higher level concepts, I think Fortino et al (2015) do a decent thing, which is to take into account some viewpoints that domains have on IoT. For example Fortino et al specifically identify 'physical properties' of smart objects to be important in some cases. So I think it might be worth paying attention to the viewpoints surrounding IoT, to get some idea about how IoT might develop; and then making sure the metamodel can handle any envisioned developments. Even if that means taking fictional ideas seriously enough to model.

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. 1297-1302. DOI: 10.1109/SMC.2015.231. [Accessed 24 March 2023]

Links

NAFv4: **https://www.nato.int/nato_static_fl2014/assets/pdf/2021/1/pdf/NAFv4_2020.09.pdf**

ISO/IEC/IEEE 42020:2019: **https://www.iso.org/standard/68982.html**

ISO/IEC/IEEE 15288:2015: **https://www.iso.org/standard/63711.html**

My Initial Post / Peer C Response

Thank you for your post.

I agree completely and think it is highly insightful to regard the fact that metamodeling provides a ‘standardised’ approach to the development of software as a strength. Without a standard or at least expectation regarding what the planning phase should produce, the goals of the modelling stage of software development can become very convoluted and not directed and focussed towards a goal and clear outcome. Furthermore, and as mentioned, prioritisation of time efficiency in this aspect of the development process is highly valuable as lack of time is something that plagues all development projects.

Finally, I agree that metamodeling techniques for IoT will become likely outdated as technology advances and the complexity of their planning requirements advance with them. Although, I view this as a great opportunity to continue to ensure that the standard processes of metamodeling are refined and themselves advanced in a way that remains clear for those who use them. In effect, when the metamodeling techniques must be advanced, so must they be reviewed and perfected.

Reference:
Fortino, G, Guerrieri, A & Savaglio C 2015, 'Towards a development methodology for smart object-oriented IoT systems: A metamodel approach', IEEE International Conference on Systems, Man and Cybernetics (SMC), University of Calabria, October 2015.

Peer A Initial Post

The meta-model approach to modelling IoT systems can be beneficial for systems modelling. As discussed in the article, various approaches have been proposed, each of which can be applied to various problems, requirements or stages in the systems design cycle. Benefits include reusability and scalability. Meta-models can be applied to various systems and could, therefore, potentially be reused in a later project or later design stages. As is evident from the article, meta-models can be relatively simple and small high-level representations of a system or far more detailed and larger representations, meaning meta-models can be scaled according to requirements.

A drawback appears to be that, currently, the meta-model approach is not standardized. This could lead to confusion across teams or within teams. However, this may change in the future, and, based on the article, it is definitely possible to apply meta-models to all system design stages. If it indeed changes, this could become an advantage of this approach. Additionally, users of this approach must be aware that high-level designs might miss out on important aspects, and extra attention might have to be paid to the design in later stages.

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. 1297-1302. DOI: 10.1109/SMC.2015.231.

Peer A Initial Post / My Response

Hi,

I definitely agree that metamodels can help achieve reusability, and scalability within a domain. Metamodels provide consistency across models (Kent, 2002), by underpinning a shared modelling language. If the language is good, it can be reused an unlimited amount of times, to make as big projects as necessary. A classic example of the benefits of a metamodel, can be seen through the success of UML, which is based on a MOF metamodel (OMG, 2017).

I also agree that not having a standardised metamodel for the IoT is a drawback. **Should IoT developers stick to modelling with UML, or is it worth having a new metamodel, and new modelling language devoted to IoT?**

In the case of the IoT, I don't believe it's enough to have a good metamodel to make it better though, because as you say, they can miss out on important aspects. For example, it is vital to have shared communication protocols that can facilitate the needs of the IoT too. The IoT is just a more complicated version of the internet, which is dependent on it's protocol suite. A metamodel provides software modelling freedom, but I don't think it addresses all of the infrastructure challenges in actualising the IoT.

References

- Kent, S. (2002) 'Model Driven Engineering', in: Butler, M., Petre, L., Sere, K. (2002) *Integrated Formal Methods*. Heidelberg: Springer. 286-298. Available from: https://doi.org/10.1007/3-540-47884-1_16
- OMG (2017). About the Unified Modelling Language Specification Version 2.5.1. Available from: https://www.omg.org/spec/UML/2.5.1/About-UM

Peer B Initial Post

Some of the strengths of a metamodel that supports the object-oriented design of the IoT are reusability, consistency, and modularity, simplifying the design process and are factors which can save considerable development time. However, it also presents challenges, including complexity, limited adaptability and learning overhead, which could pose as a barrier to adoption.

While a metamodel can provide a base approach to IoT design, developers must balance its benefits with potential drawbacks, considering factors such as project-specific requirements, adaptability, and the flexibility to explore alternative solutions.

Peer B Initial Post / My Response

Hi,
I think the learning overhead can also be a barrier to adoption. Proving a metamodel is good enough for a whole domain such as IoT sounds almost impossible, so it probably wont be attempted by many people. Are we doomed to wait for decades of research before we finally have something everyone can agree on, or **is there some way to speed up the research and adoption of a good enough metamodel for IoT?**

Peer B Initial Post / My Response / Peer B Response

Hi,

I wouldn't say we are completely doomed, but it's definitely going to be quite a long time before a method, or approach that everyone agrees on is made the standard. **An approach to speed up the research and adoption of a good enough metamodel for IoT, a large technological corporation would have to take the mantle** and start publishing tools and standards as to how they develop their own metamodels in-house, I feel that this way would encourage more people to adopt such methods and hopefully, with time it becomes a standard.

Peer C Initial Post

The Internet of Things (IoT) describes real world objects that participate in the internet and therefore, must be globally networked, discovered and exploited. Smart Objects (SOs) which are physical objects that are actual implementations of IoT, are strengthened by improvements in processing, communication, sensing and actuation functionality. To fully exploit an SO's potential, Fortino et al. (2015) suggest modelling of the system's properties must occur at different levels of abstraction. They attempt utilising metamodels. Different versions of metamodeling map SO basic features and high-level interactions.

Of course, metamodeling has strengths and weaknesses and these directly relate to the successful implementation of improved SO functionality. These will be discussed. Primarily, the strength of a metamodel lies in its reduction of the computational burden associated with computationally demanding analyses of simulation models. Thus, they are comparatively quick to develop, simple to derive, extremely flexible and more cost efficient. Furthermore, they are well suited to deterministic applications such as SOs. And finally, are useful in 'linearizing' otherwise highly complex conceptualisations.

Weaknesses of metamodeling as they relate to SOs must be considered. Firstly, depending on the complexity of the SO conceptualisation or application, 'linearization' can be difficult to extract; specialist expertise may be required depending. Furthermore, the adequacy of the model is determined solely by individual conclusions. And finally, at this time there exists little suitable software for expediting and making efficient the modelling process.

Reference:

Fortino, G, Guerrieri, A & Savaglio C 2015, 'Towards a development methodology for smart object-oriented IoT systems: A metamodel approach', *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, University of Calabria, October 2015.

Peer C Initial Post / My Response

Hi,

I think it's a great point that a good metamodel can potentially simplify complex systems. There are certain metaclasses of the MOF metamodel

(OMG, 2017) that seem essential for modelling, such as 'attribute' and 'class'. I can't imagine there would be any need to represent something that doesn't have a title (class) and attribute, so those metaclasses seem essential. I guess new metaclasses could provide convenience too though. **Can you think of any more simple concepts, that would make modelling a complex system like IoT software, simpler; or are Fortino's suggestions simple enough for any IoT developer to use?**

References

OMG (2017). About the Unified Modelling Language Specification Version 2.5.1. Available from: https://www.omg.org/spec/UML/2.5.1/About-UML

Peer D Initial Post

The main advantage of the metamodels is their approach in the way they tackle the IoT system implementation. As stated in the article, these kinds of models offer pretty good modeling from abstract to detailed system design, hence utilizing all aspects of smart objects, including the part where software development comes into play. However, as stated by the author, this may also be a drawback, as no well-formalized methodology has yet been adopted, moreover the use of software engineering in IoT devices might still be in its early ages. Taking this into account, it is therefore important to standardize the design of metamodels for particular IoT systems. As for comparison, the UML diagrams are well standardized, however in literature we can still find different modeling techniques and terms that are being used.

Overall, metamodels have their strengths in designing complex industry-specific systems and can be applied by multiple stakeholders in similar industry sectors, thus holding the value of reusability.

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. 1297-1302. DOI: 10.1109/SMC.2015.231. [Accessed 24 March 2023]

Peer D Initial Post / My Response

Hi,

Thanks for the well thought out post. I agree that standardization is a good thing,

I'm wondering, would you say there is anything specifically wrong with UML for designing IoT software? I mean**, is there any reason that UML must be replaced for IoT software modelling?** I don't think there is a reason, but I think perhaps a new modelling language that has higher-level entities might be easier to use. Maybe entities that are too high level can be a problem too?

Peer D Initial Post / My Response / Peer D Response

Hi,

Thank you for the question. I wouldn't say, there is anything wrong with the use of UML for IoT software modeling, as the **UMLs are already well established**. Moreover I would agree with you regarding new modeling language specific for IoT, as the field of smart objects by itself is likely different in contrast to pure software development, hence higher-level entity would probably enhance the development quality. However, on the other hand, entities with higher-level may present a problem, as an 'additional redundant layer', if the objectives at the beginning of a project are not properly set.

Peer E Initial Post

The metamodel approach to designing IoT systems can be beneficial. The article discusses the varying level of uses and abstraction from starting at a high level and supporting different stages of the lifecycle to more detailed and granular designs. From that aspect, key strengths would include reusability and scalability as well as different model types used at varying stages of the design if requirements and analysis are clear at the start.

It does however introduce several drawbacks or perceived weaknesses. Mainly the fact that there is no published standard for metamodel design

so may be challenging to adopt. Additionally, such an approach could lead to varying levels of complexity and risks of becoming redundant as designs and solutions become more agile in their delivery. Investing heavily in a detailed Metamodel could prove costly when a significant change occurs if trying to design too much upfront. The authors conclude that subjects such as security are not tackled. This is a prime example of how a critical aspect of design could be missed or not considered and therefore prove costly to refactor designs, especially with cloud-based IoT, disparate systems and authentication methods.

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. 1297-1302. DOI: 10.1109/SMC.2015.231.

Peer E Initial Post / My Response

Hi, thanks for the post.

I agree that investing in a metamodel might be costly if significant changes occur to the structure of the IoT. I'm wondering if you have any insight as to what sort of standards would be necessary to coordinate the development of the IoT?

Once standards are agreed upon by an authority, would a well-documented, easy-to-use metamodel/model be capable of pushing those standards, to other developers?