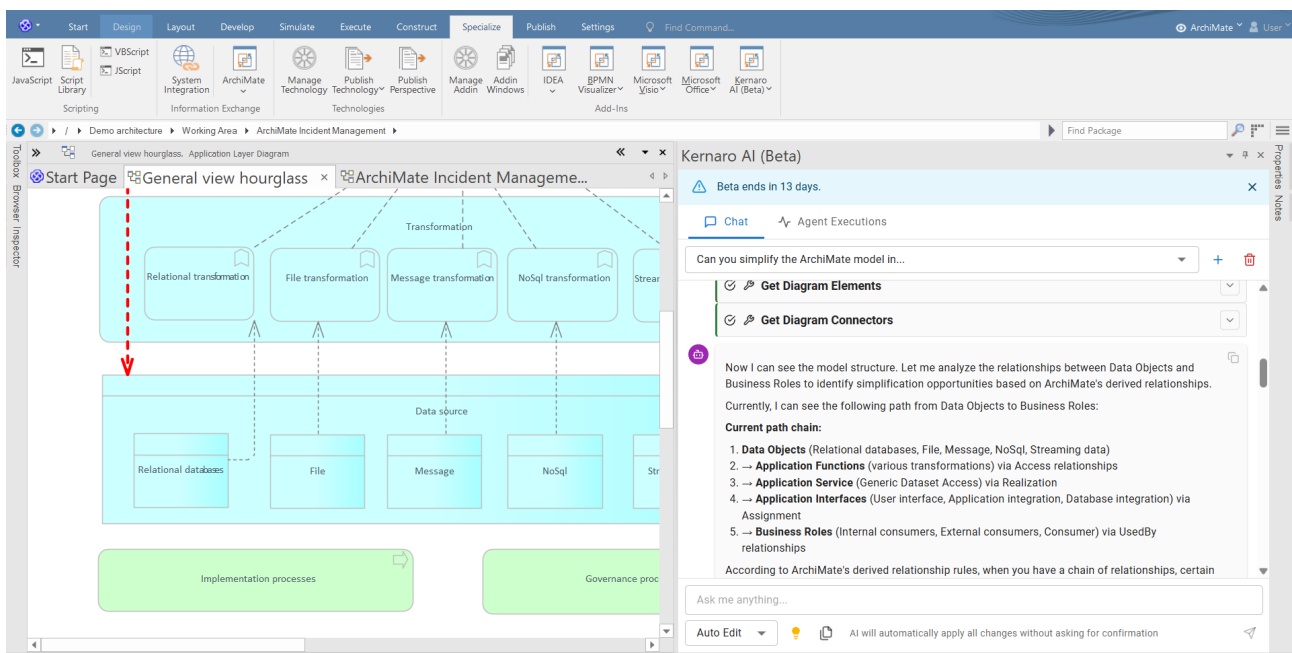


AI addon Kernaro for Sparx Enterprise Architect



By Bert Dingemans

<https://data-docent.nl>

April 2026

Introduction

Artificial Intelligence based on Large Language Models is everywhere around us. There are all kinds of tools and techniques introduced. Also for us as modelers there are a lot of things happening. In my daily work I extensively use Claude for evaluation and generating all kinds of models. My models are mainly Enterprise Architecture models in ArchiMate and datamodel in UML Class and Database models.

Recently Sparx Systems introduced Kernaro an AddOn for Sparx Enterprise for using Generative AI for models in Sparx Enterprise Architect. This is an interesting AddOn for us as modelers so I decided to do a number of tests on existing models in a repository and create a number of new models. I received a betatest version of Kernaro and did some tests. I am impressed by the results of the AddOn.

Setup

The setup I used for the evaluation of the Kernaro AddOn is relatively simple. First I created a model that connects to Claude from Anthropic. For that I have to create an API key in the platform of Claude, buy a small amount of credits and this is sufficient for start using Kernaro in Sparx. I did only test the Chat function but this is more than sufficient to get use AI in Sparx with Kernaro.

Tests

In this chapter I will do a number of tests on ArchiMate and datamodels. The setup of every test is the same I formulate a question (in italics) and describe what the outcome is of the Kernaro results. This can either be some text or a diagram or package with a model as result.

In the tests I used Incident Management as a small domain for creating ArchiMate and datamodels. Reason to choose for incident management is that it is less common than relationship management or order entry.

Review an ArchiMate model

In the package BigData assistant there is an ArchiMate model. Can you evaluate this model for me. Is the ArchiMate model correct and are ArchiMate viewpoints applied in the model.

Based on this question Kernaro does a thorough evaluation of the model present and comes with the following results. The results are informative for evaluating an ArchiMate model for example for reviewing a model based on correctness of the metamodel. This will reduce the workload of a repository custodian or model manager enormously.

Based on my thorough analysis of the ArchiMate ArchI model in the "BigData Assistant" package, I can now provide you with a comprehensive evaluation.

ArchiMate Model Evaluation - BigData Assistant Package

Summary

The model contains **valid ArchiMate elements and relationships** but has **significant issues with viewpoint application and some modeling inconsistencies**.

✔ What's Correct:

1. **Valid ArchiMate Elements** - The model uses proper ArchiMate 2.x/3.x stereotyped elements:

- **Application Layer:** ApplicationFunction, ApplicationService, ApplicationInterface
- **Business Layer:** BusinessRole, BusinessProcess
- **Motivation Layer:** Requirement, Constraint
- **Data Layer:** DataObject

2. **Correct ArchiMate Relationships:**

- Access relationships (functions accessing data objects)
- Realization relationships (functions realizing services)
- Aggregation relationships (composing interfaces and roles)
- Association relationships (linking constraints and requirements to services)
- UsedBy/Serving relationships (interfaces serving consumers)

3. **Good Modeling Practices:**

- Clear separation of concerns (data sources, transformation, standardized data, interfaces, consumers)
 - Logical layering following the "hourglass" pattern
 - Proper use of aggregation for composition (e.g., specific interface types aggregating to generic interface)
 - Appropriate stereotyping of elements
-

✘ Issues Identified:

1. **ArchiMate Viewpoints NOT Properly Applied**

Critical Issue: The diagrams are marked as ArchiMate diagram types but are NOT using proper ArchiMate viewpoints:

- **"Big Data Overview"** (Main package diagram):
 - MetaType: ArchiMate3::Application
 - Problem: Contains only navigation cell text elements, no actual ArchiMate content
- **"General view hourglass"**:
 - MetaType: Archimate2::Application
 - ✔ Contains proper Application Layer elements
 - ⚠ But also mixes Business Layer (BusinessProcess, BusinessRole) and Motivation elements (Requirement, Constraint)
 - Should be: **Application Cooperation or Application Usage viewpoint** (allows business roles accessing application services)
- **"Consumers overview"**:
 - MetaType: Archimate2::Business
 - ✔ Correctly focuses on Business Layer (BusinessRole)
 - ✔ Proper use of Business Layer viewpoint

•"Detailview interfaces":

- MetaType: Archimate2::Application
- Correctly focuses on ApplicationInterface elements
- Appropriate for Application Structure or Application Usage viewpoint

2. Inconsistent Connector Stereotypes

Some connectors use EA base types instead of ArchiMate stereotypes:

- Dependency type used with ArchiMate_Access stereotype
- Aggregation type used with ArchiMate_Aggregation stereotype
- Association type used with ArchiMate_Association stereotype

Note: This is technically correct (EA approach), but ensure consistency across the model.

3. Mixed Element Types

The "General view hourglass" diagram mixes:

- Application Layer (functions, services, interfaces)
- Business Layer (business processes, business roles)
- Motivation Layer (requirements, constraints)
- Data Layer (data objects)

Recommendation: Consider splitting into separate viewpoints:

- Application Cooperation Viewpoint:** Focus on application services and functions
- Application Usage Viewpoint:** Show how business roles use application services
- Motivation Viewpoint:** Link requirements/constraints to application elements

4. Navigation Elements in Main Package

The main "BigData Assistant" package contains 12 Text elements with "NavigationCell" stereotype - these are not ArchiMate elements and should ideally be in a separate navigation/overview diagram or removed.

ArchiMate Viewpoint Recommendations:

For a Big Data architecture, consider these standard ArchiMate viewpoints:

1.Application Cooperation Viewpoint

- Shows: Transformation functions, Generic Dataset Access service, Data access interfaces
- Focus: How application functions work together

2.Application Usage Viewpoint

- Shows: Consumers (business roles) using application interfaces
- Includes: User interface, Database integration, Application integration

3.Information Structure Viewpoint

- Shows: Data objects (Relational databases, File, Message, NoSql, Streaming data, Standardized object)
- Focus: Data structures and relationships

4.Motivation Viewpoint

- Shows: Data Qualities (requirement), Connection requirements (constraint)
- Links to services/interfaces they constrain

5.Business Process Cooperation Viewpoint

- Shows: Governance processes, Implementation processes
 - Links to the Transformation function
-

Recommendations:

1. Define **explicit viewpoints** for each diagram based on ArchiMate standard viewpoints
 2. Remove **navigation cells** from the main package or consolidate into a single navigation diagram
 3. Split the "General view hourglass" into 2-3 focused viewpoints:
 - One for application layer
 - One for business-to-application interactions
 - One for requirements/constraints
 4. Document **viewpoint purpose** in diagram notes to clarify stakeholder concerns addressed
 5. Ensure **consistency** in connector stereotyping throughout the model
 6. Consider **upgrading** fully to ArchiMate 3.x (currently mixing ArchiMate 2 and 3 diagram types)
-

Overall Assessment:

| Aspect | Status | Score | |-----|-----|-----| | **ArchiMate Element Types** |  Correct | 9/10 |
| **ArchiMate Relationships** |  Correct | 8/10 | | **ArchiMate Viewpoints** |  Not Applied | 3/10 | | **Model Consistency** |  Partial | 6/10 | | **Layering & Structure** |  Good | 8/10 |

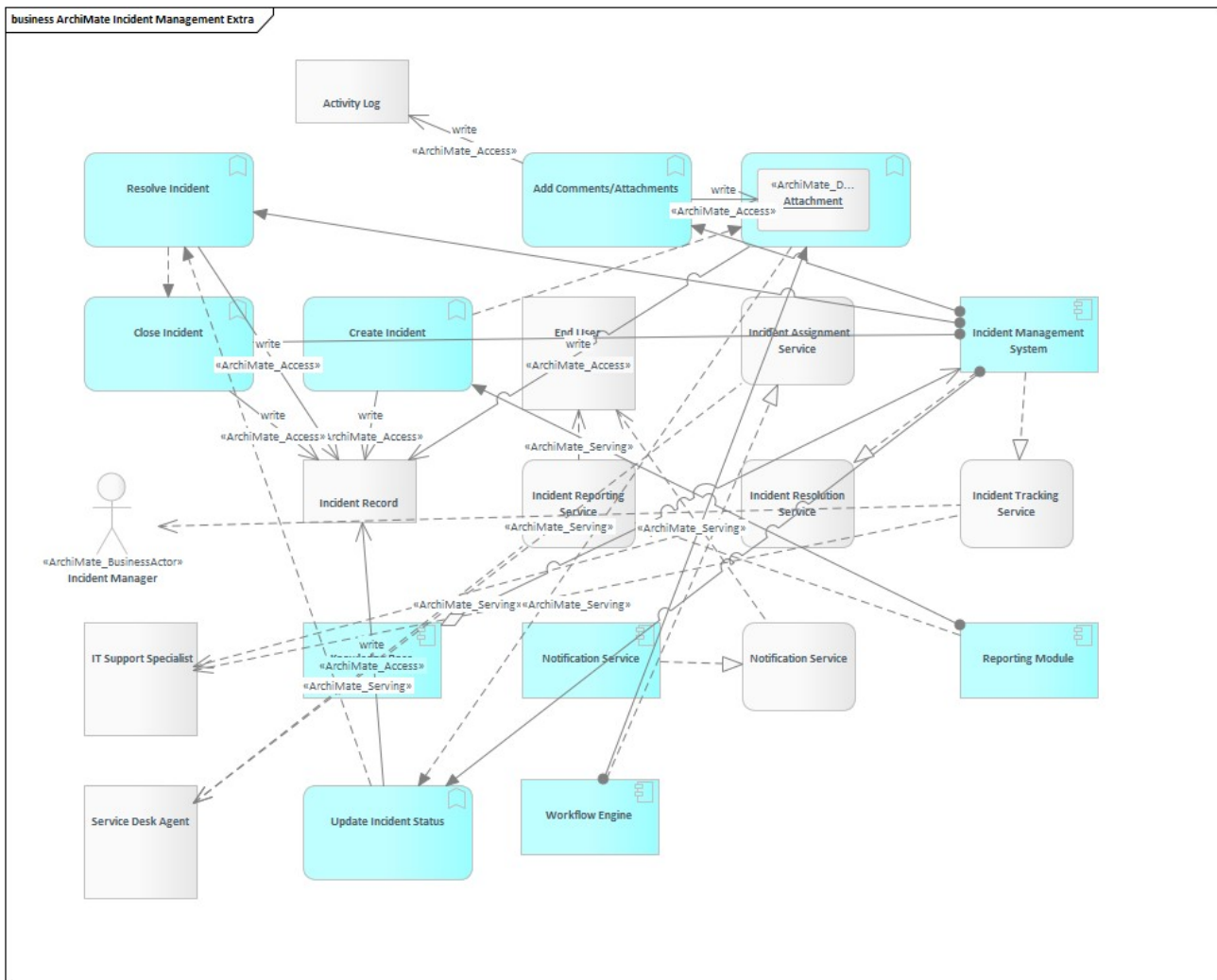
Overall Rating: 6.8/10

The model uses ArchiMate correctly at the **element and relationship level**, but **fails to apply proper viewpoints** which is essential for stakeholder communication and ArchiMate compliance. The content is architecturally sound, but needs better organization into proper viewpoint-based diagrams.

Create a simple ArchiMate model

Can you generate an ArchiMate model and diagram based on the application usage viewpoint for the incident management process. Put the results in the package ArchiMate Incident Management.

Below you see the diagram Kernaro generates based on this question. It is impressive how extensive the model is that is created.



The elements and relationships are based on the viewpoint as requested. The application components and application functions are related in a correct manner and also the flows between the functions are modeled correctly. Nice is that the access relations also include write as a name.

What is missing in this diagram is that there are no business layer elements used in the model like business processes and eventually business objects. These are part of the application usage viewpoint.

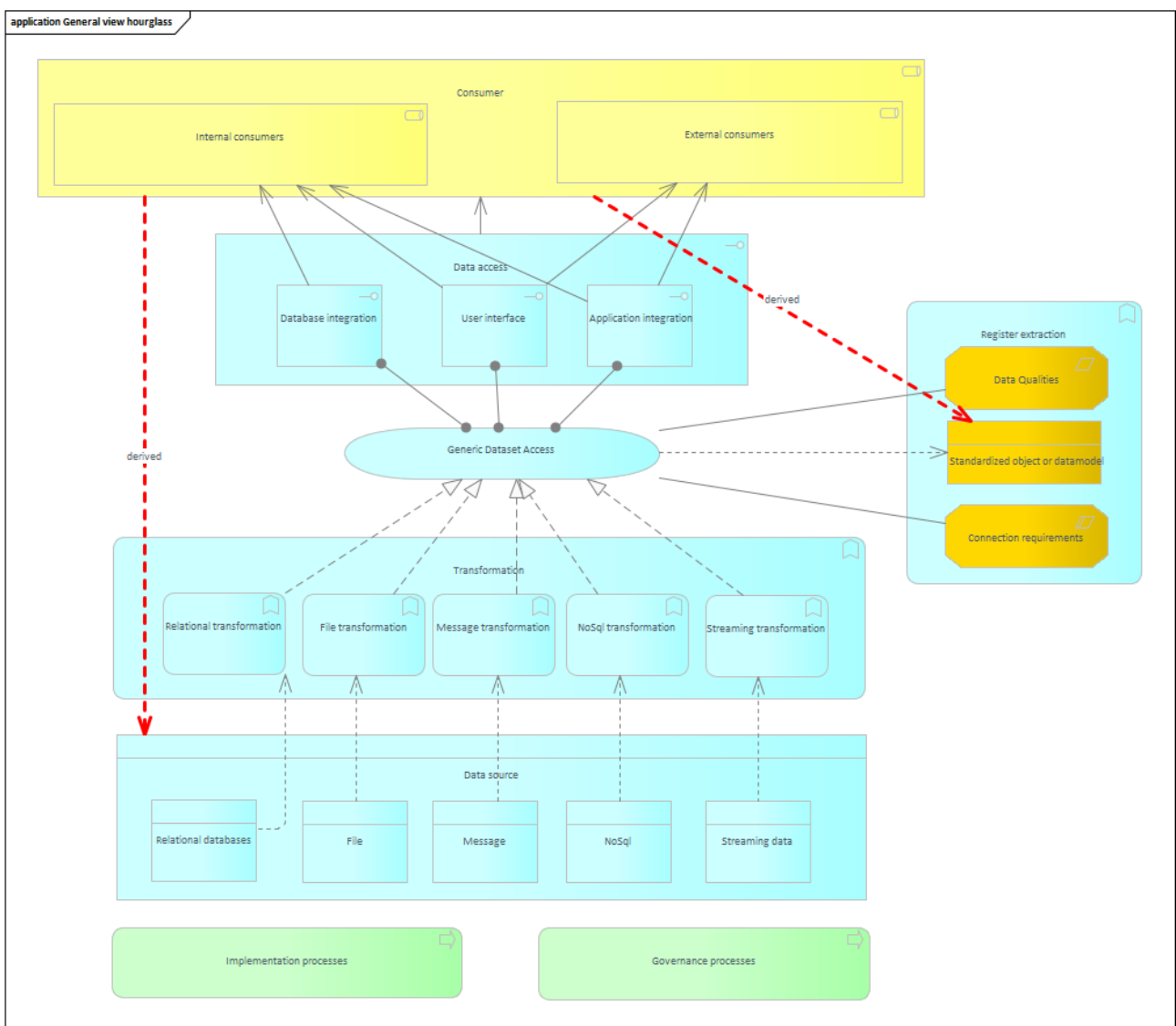
Unfortunately the stereotyping is not fully correct. However this can be based on the internal configuration of Sparx with objecttypes and stereotypes.

Simplify the ArchiMate model based on the derived relationships

Can you simplify the ArchiMate model in the diagram General View Hourglass based on the derived relationships in the metamodel of ArchiMate. Simplify between Data Objects and Business Roles.

Below you see the diagram we used as starting point for creating a simplified model, it includes the simplified model in which I added the coloring of the new derived relationships. The expected result is that there is an access relationship between the various data objects and the business roles.

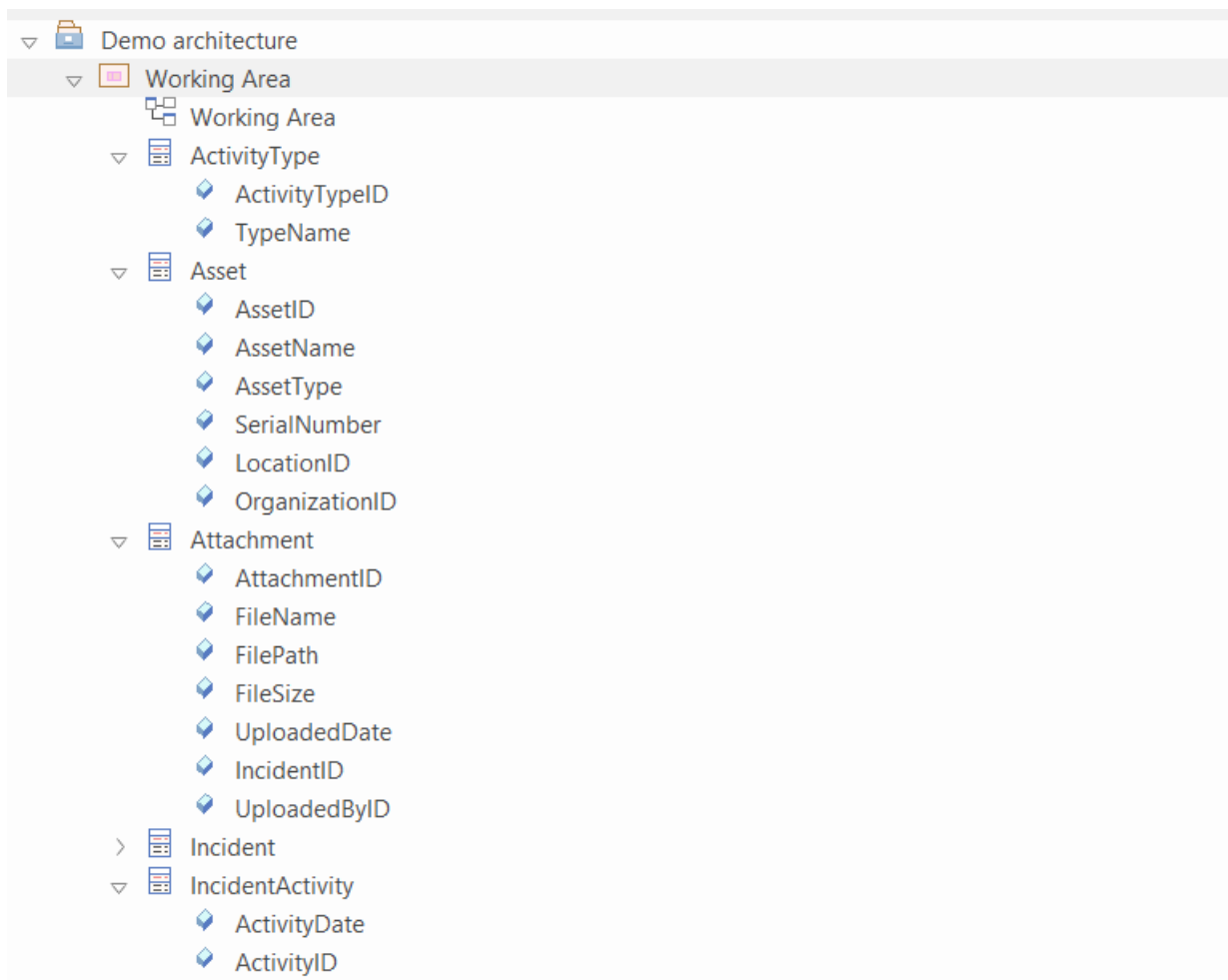
This results nicely includes the aggregation of the data object elements and only creates an access relationship to the aggregated data objects. The direction of the access relationship is not included in the model which is a point of discussion. However impressive how the model is modified based on the metamodel of ArchiMate.



Create a logical data model for incident management

Create a logical datamodel in the package working area of incident management. Create the entities and the attributes

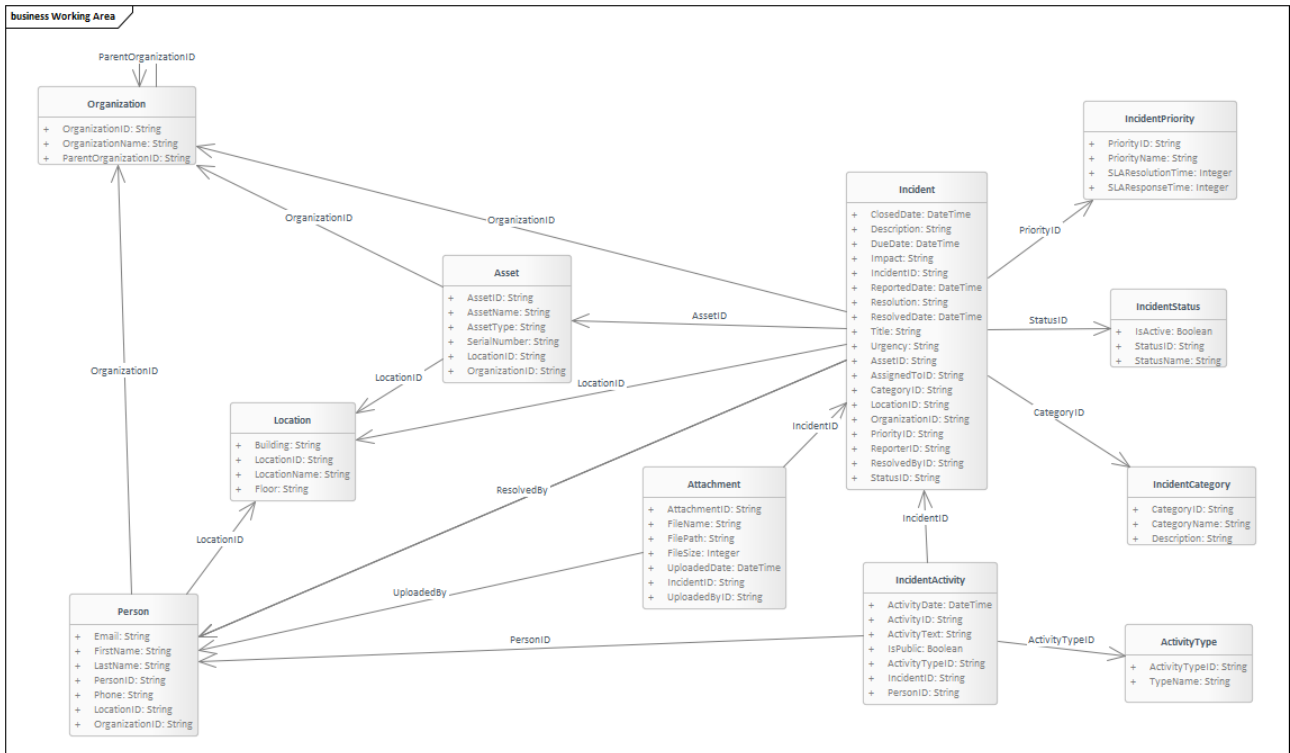
Below you see that based on the question asked Kernaro generated a logical datamodel in a UML setting. It includes datatypes based on VB.Net as set in the Sparx settings. Furthermore it uses a nice naming convention. Some datatypes are defined as strings but could be numeric also. But it is a good starting point. Impressive is the datamodel used for incident management.



Create relationships and foreign keys for the logical data model

In the package working area there is a logical datamodel can you create the relationships based on the naming conventions of the attributes.

This is a nice logical datamodel that is created by Sparx and Kernaro. It extends the model first with foreign keys in the other elements as attributes. It applies a naming convention by itself. Furthermore the relationships are created accordingly. Based on the model it would be nice that the cardinalities of the relationships are also created. But as a starting point this is a good logical datamodel.



Evaluation and suggestions

Based on this small evaluation of the Kernaro AddOn for Sparx Enterprise Architect I am impressed of the functionality and the models that are created and evaluated by the AddOn in combination with Claude. The models are maybe not fully correct but also the questions asked can be limited in this.

The artifacts created will not replate the work of a modeler (yet) but it will reduce the workload of modelers enourmously. Very interesting is that it can validate created models based on a metamodel of a modeling language.

There are some suggestions for the Kernaro addon:

- Error messages are some times hard to understand and not informative about what goes wrong.
- It would be nice that the chat function is available as a context menu for packages, diagrams, elements and maybe even connectors. That will reduce the questions setup for the modelers.
- A feature request would be that the messages created in the chat can be exported to a document for documenting purposes. Maybe even more elegant is that the messages are added to a custom document artifact in the repository you are evaluating.

Summarizing: I am really impressed of the work the team of Kernaro delivered. This is a powerful AddOn that will be really helpfull for every modeler using Sparx Enterprise Architect. The integration of Sparx and Claude is reducing the workload for every modeler.

More information about Kernaro can be found at <https://kernaro.sparxsystems.com/kernaro-ai-hub/>

About the author



Bert Dingemans is a trainer in the field of data architecture, data management and Big Data. He has a passion for modeling, modeling tools and the effective use of automated tools to effectively use models in practice. More whitepapers can be found at <https://data-docent.nl> . Bert can be reached by email at bert@data-docent.nl