

SYSEN 6150: Model Based Systems Engineering

Subsystem Matrix & Interface Diagram

Nick Kunz [NetID: [nhk37](#)] nhk37@cornell.edu

October 28, 2022

Subsystem Matrix

The following is an example of a subsystem matrix for the transit prediction system as the System of Interest (SOI).

Data Collection System	Data Parser and Mapping System	Data Storage System	Analytics and Modeling System	Geographic Information System
The system shall authenticate with API end point and key.	The system shall conduct identification and object mapping	The system shall conduct data persistence & storage.	The system shall automate analyses and interpretability.	The system shall conduct spatio-temporal analyses.
The system shall send get requests for data retrieval.	The system shall filter for id inconsistencies	The system shall store service logging.	The system shall conduct model training & validation when required.	The system shall conduct geoprocessing when required.
The system shall handle unresponsive and stale requests.	The system shall conduct data transformations.	The system shall be able to query data.	The system shall detect outliers.	The system shall geographically visualize GIS data.

Interface Diagrams

The SOI often has several important subsystems, such as those previously mentioned in the Subsystem Matrix above. The layers between those subsystems are equally important, as they bind them to their respective interfaces. The interfaces between subsystems are critical to the proper functioning of the broader SOI, where the goal is to design interfaces between subsystems, such that all components appropriately integrate with each other and that the broader SOI functions as intended.

In the case where the Interface Diagrams are applied to the given transit prediction system as the SOI, the Interface Diagrams are perhaps the most obviously useful in the *Data Collection System*. The *Data Collection System* necessarily requires a connection to the source Application Programming Interface (API). Furthermore, it needs to send and receive data through the same interface, as well as appropriately handle events where those requests for data are unsuccessful.

Another application of the Interface Diagram is the layer between the *Data Parser and Mapping System* and the *Data Storage System*. Assuming the data from the API has been successfully retrieved and processed, the SOI requires that the data be persisted. In this context, the *Data Parser and Mapping System* needs to write to a given database, which occurs at the interface of the *Data Storage System*. Furthermore, that all other interface events be logged and stored in the *Data Storage System*.

In the two previously mentioned examples, the interfaces generally only require integration between two other subsystems (with the main exception of logging). However, there is such a case where a subsystem requires an interface between four other subsystems. The *Analytics and Modeling System* interfaces directly with all other subsystems other than the *Data Collection System*. It is important to highlight here that this single subsystem requires four distinct sets of interfaces.

Interfaces and the subsystems they connect are important considerations when designing any SOI. The proper functioning of each subsystem and the broader SOI relies on them, which is why Interface Diagrams are important when designing such a system. Interface Diagrams aid in outlining the relationships and requirements between subsystems. They can also help to describe other systems beyond those described within the SOI and their respective interface requirements.