Annotation guidelines for Problem Diagrams dataset

1. Class
   1. Entities

* **Machine Domain**: is the system or software that you want to design and build. Here are some examples: the sun search control system, the smart home system. Please note if “the system” appear in the requirements description, we should label it. Because “the system” often has relation with other entities. If we don’t label it, many relations can’t be labeled. (Machine Domain is software)
* **Physical Device**: is device in the real world, which can be used to get/send/receive data/info. Here are some examples: the sensor, the actuator and so on.
* **Environment Entity**: is entities in the real world. Here are some examples: the patient, the sun, the old and so on.
* **Designed Domain**: is manually designed domains that already exist for the develop software to be developed. Their properties are artificially designed or prescribed. Here are some examples: database, a magnetic stripe card, a tap, hard disk, floppy, and **printed output.**
* **Shared Phenomena:** is a set of shared events, states and values between the connected entities. Here are some examples: send data, notification.
* **Requirements:** is the aim of the system to be developed. Here are some examples: to monitor sun, the control the home, to monitor the patient and so on.
  1. Interactions
* **Interface:** is an interface of shared phenomena between the connected entities. For example, the smart medical system can send notification to the nurse’s station. It contains a ternary relation <The smart medical system, the nurse’s station, the notification>. Here we divided it into two interface relations. The first interface relation is , it is the output direction of the message or the data flowing. The second interface relation is , it’s the input direction of the message or data flowing. The interface relation exists between Machine Domain and Shared Phenomena, between Physical Device and Shared Phenomena, between Environment Entity and Shared Phenomena.

Note: if there is no data/event, we just connect two domain entities.

* **Requirements reference**: is reference relation between Requirements Domain and other entities. For example, the smart medical system need to get the health condition of the patient in order to monitor the patient. It contains a ternary relation. It contains a ternary relation <The patient, to monitor the patient, the health condition>. Here we divided it into two requirements reference relations. The first relation is <The patient, the health condition>. The second relation is <the health condition, to monitor the patient>. The requirements reference relation exists between Requirements Domain and other entities.

Note: if there is no data/event, we just connect two domain entities.

* **Requirements constraint**: is constrain relation between Requirements Domain and other entities. It means the Requirements Domains does not just refer to the phenomena but constrains them. For example, the smart medical system needs to medical watch to get the health condition of the patient in order to monitor the patient. It contains a ternary relation<The medical watch, to monitor the patient, the health condition>. Here we also divided it into two requirements constrain relations. The first relation is <The medical watch, the health condition>. The second relation is <the health condition, to monitor patient>. The requirements reference relation exists between Requirements Domain and other entities.

Note: if there is no data/event, we just connect two domain entities.

**Please Note**: Requirement references means that the requirement phenomena also occur without the software system to be developed, while requirement constraints means that the requirement phenomena must occur after the software system to be developed.