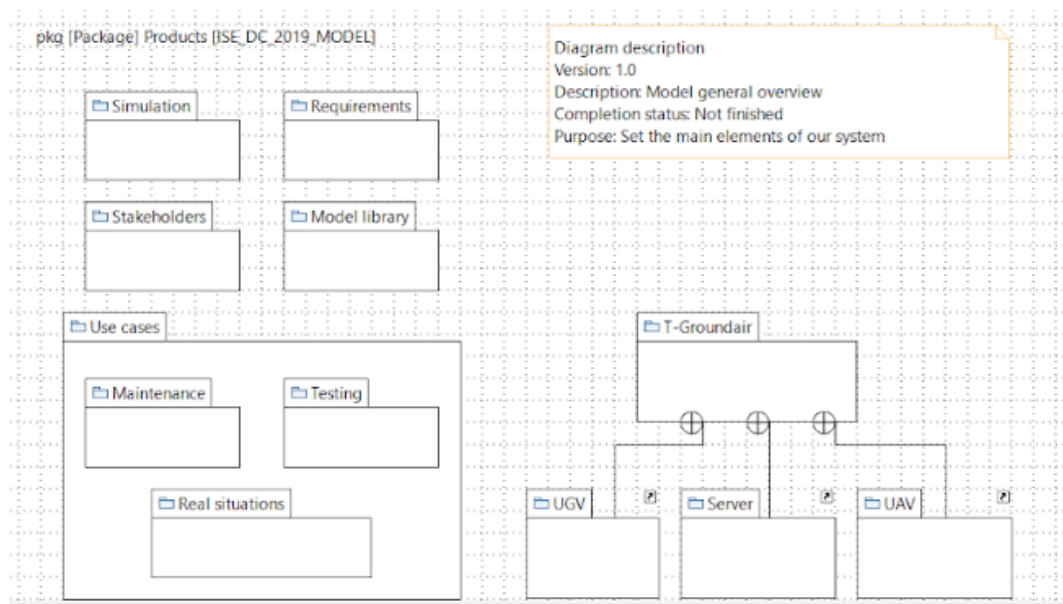


Modeling Guidelines

The Project's SysML Model is a powerful tool which is intended to be used by all members of ISE 2019 Team. Because of that, following some format rules and a common philosophy is a real necessity in order to develop a useful model. In this document, which is subject to any kind of suggestion or change you may come up with, we will try to set these common lines and give you some ideas for the development of your contributions.

1. Format key facts

As you will see, our model is composed by many different diagrams. In order to preserve an organized structure, it is important to meet the following format requirements in every diagram you create.



When opening the model, this is the first screen you will see (at least, when this version was written - 1-2-2020). As shown at the picture above, every diagram which is part of the model must include the diagram header, written as a *text* at the *top left corner* of the page, and the diagram description, written as a *note* at the *top right corner*.

Diagram header: it follows the SysML standard format, containing the information which is described below.

diagram kind [model element type] model element name [diagram name]

- Diagram kind: in this context, some of the most used codes could be **act** (activity diagram), **bdd** (block definition diagram), **ibd** (internal block diagram), **pkg** (package diagram), **rqt** (requirement diagram), and **uc** (use case diagram).
- Model element type: depending on the diagram, elements such as activity, block, package, model library or requirement might be used and mentioned at this field.
- Model element and diagram names: they can be freely chosen depending on the information described at each specific diagram.

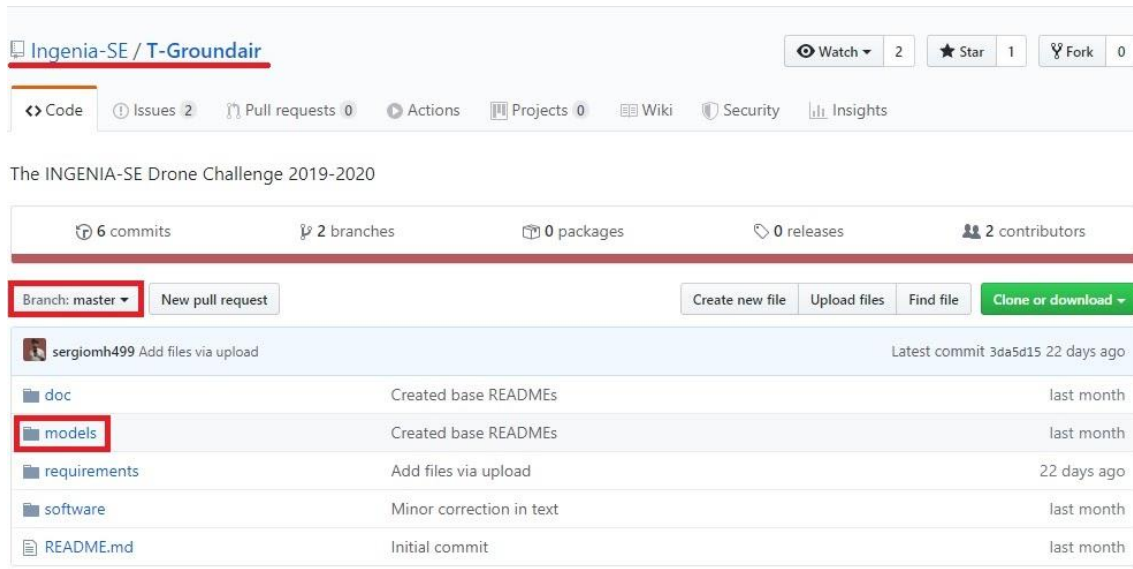
Diagram description: it includes some information that we consider useful for the general understanding of the model, containing also some extra fields which are usually part of SysML viewpoints. Trying to ease the diagrams construction and reading, we have added some of their points here.

- Version: numbered as *a.b*, where *a* is updated when a major change is made, and *b* is incremented in case of a minor update.
- Description: main ideas about the diagram's content.
- Completion status: *finished* / *not finished* / *under discussion* (see Section 2 - Version Control).
- Purpose: this is the place for explaining the diagram's goals. Trying to answer questions such as *Which extra valuable information does this diagram give?* or *Who are the stakeholders of this diagram?* might help.

2. Version control

In order to update the model, we will follow a simple procedure to make sure everything is correct before the change is made.

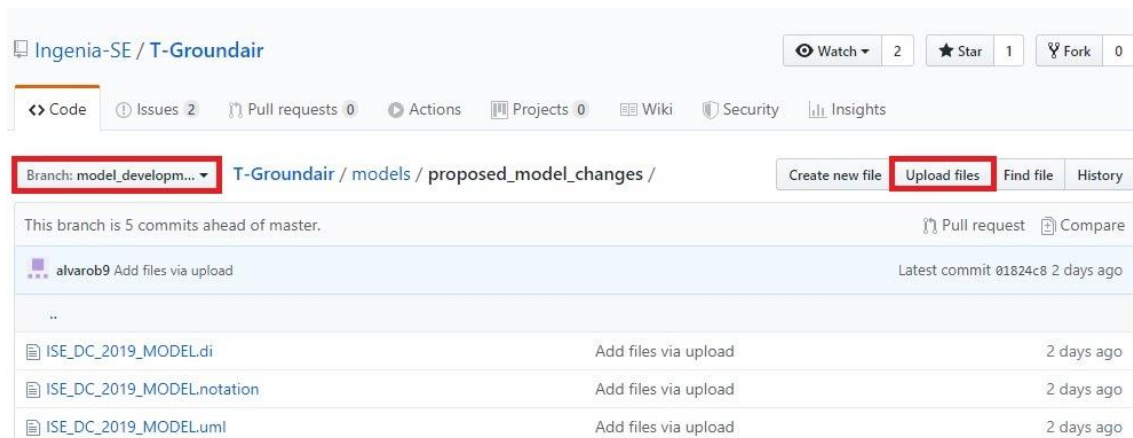
Firstly, anyone who wants to make a change shall download the latest updated version of the model. In order to find this file, the students shall enter the Ingenia-SE GitHub. Once there, click on the "T-Groundair" repository and then in the "models" folder. For this step, make sure you are in the "master" branch.



The screenshot shows the GitHub repository page for 'Ingenia-SE / T-Groundair'. The repository has 6 commits, 2 branches, 0 packages, 0 releases, and 2 contributors. The 'master' branch is selected. A table of files is displayed:

File	Commit Message	Time
doc	Created base READMEs	last month
models	Created base READMEs	last month
requirements	Add files via upload	22 days ago
software	Minor correction in text	last month
README.md	Initial commit	last month

When the modifications are included, in order to upload the new version of the model, you shall select the “model_development” branch; after that, click on the “models” folder and then click on the “proposed_model_changes” folder. Once there you can upload the new files by clicking on “Upload files” and dragging them into your browser window.



The screenshot shows the GitHub repository page for 'Ingenia-SE / T-Groundair' with the 'model_development' branch selected. The repository is 5 commits ahead of master. A table of files is displayed:

File	Commit Message	Time
ISE_DC_2019_MODEL.di	Add files via upload	2 days ago
ISE_DC_2019_MODEL.notation	Add files via upload	2 days ago
ISE_DC_2019_MODEL.uml	Add files via upload	2 days ago

Finally, once these files are verified by Modeling Management and Configuration Management, they will be uploaded to the “master” branch, so that they can be downloaded for subsequent updates.

3. Structure notes

At the beginning, the packages of our first model are mostly empty. This is made for giving each working group enough freedom for designing the diagram structure that they prefer depending on their assigned tasks. In this Section, we include some notes for helping the model's structure understanding and its development.

- **ISE_DC_2019_MODEL:** as you have seen in the image shown at Section 1, this is the main screen of our model. It includes some elements that we would like to describe with further detail:
 - **Model library:** this package is thought for containing structures which might be used repeatedly in other different packages. Its usage is optional, but we consider it as a potentially useful tool.
 - **Stakeholders:** we also think that detailing our main stakeholders is an important part of our model. Because of that, there is a specific package for that purpose. We suggest using structures such as tables to fill in that package.
 - **Use cases:** taking into consideration that we are modeling a physical system which is going to be used in real environments, it is important to describe those situations, but also maintenance and testing procedures. These packages have been created with that goal.

Finally, we would also like to outstand the relevance of using hyperlinks for moving from one diagram to another just by double clicking the chosen entity. Creating them is an easy task. We set two possible different cases:

A. Hyperlink with a non-created diagram. If you want to draw a new diagram which describes an existing structure (package, block, etc.), just *Right click on it -> Navigate -> Create hyperlinked diagram*.

B. Hyperlink with an already created diagram. Let's say you are in charge of creating the diagrams related with UGV. It's not our intention that you copy again the specific use cases and requirements inside their respective packages. We would like you to hyperlink those packages with the corresponding diagrams which will be already placed at *Requirements* package. To do so, *Right click on the package you want to hyperlink with a pre-existing diagram -> Navigate -> Modify Hyperlinks*, and then add the chosen diagram by choosing *+ -> Specific object*. Finally, press Default Hyperlinks tab and move the chosen diagram from Available hyperlinks to default hyperlinks by clicking on *->*.



Álvaro Belloso Esteban
Alejandro Gómez García

Modeling Management
V1.0 - January 2nd, 2020

We expect these main ideas to be useful for the model's development. Please, feel free to make us any suggestions of possible changes, new sections or any specific information you need about modeling. We are not experts at SysML, but we will do our best to help you. *Good luck!*

