QEES Project: Battery-Aware Scheduling in Space

Marko Groffen, Sighvatur Gìslason
Project Group 15

January 23, 2019

I. Introduction

This paper presents a short explanation of the generated UPPAAL CORA model, used to generate a transmission timing schedule for the GOMX-3 satellite. First the assumptions and abstractions of the model are compared to the previously generated model in [1]. Next, a short explenation of the model is provided. The quality of the schedule is discussed, with additional comments presented in the discussion. The quantifiable uncertainties of the model are explored and a way to model them as probabilistic choices is explained. The usage of UPPAAL STRATEGO is also investigated to generate the schedule.

II. Assumptions

There are a number of assumptions we deal with the same as [1], namely battery behavior, normalized power load and constant power income from the sun. Additional simplifications of the model compared to the paper are given by the project assignment. Uncertainties in the parameters are mentioned later in the paper.

III. Model Explanation

We recreated the model in [1] to start and worked our way into developing the model further with the heuristics, power, preheating and slewing in mind. To work along the main model, an Attitude control system (ADCS) was added to simply align with the model of aligning, waiting the correct time and releasing. The main model was set up to slew back at the end after execution of a job in accordance.

Preheating was handled with the Job Provider, accessed preheat time for each job used to wait until the start of a job, preheat, release and signal the model for availability. The Job Provider was expanded to go to an end state when we reach the end of the available time windows as well as skip a job and penalize if we missed the next upcoming time window after a previous execution.

We wanted to simplify the battery handling, so we only update at the end of a job's execution, taking into account all the power coming in and out, depending on the sun exposure. The sun was modeled as switching back and forth between insolation and eclipse, as appropriate, updating the battery after each insolation period. Finally, an automata was added to stop the simulation after a defined time. We wanted to implement some of the heuristics mentioned in [1], so the L and X-band limits were enforced in addition to always scheduling UHF jobs.

IV. Quality of the Schedule

Unfortunately, we had insufficient time to work out all the small problems with our model. There were some deadlocks or stop states which we were unable to reproduce easily and jobs kept getting skipped, the verifier unable to determine a suitable, nontrivial schedule. Adjusting the cost rates had no effect, as too many jobs were unable to execute. Since the so many jobs were skipped, the schedule is at least safe battery-wise.

V. Quantifiable Uncertainty

The model takes a lot of assumptions into consideration and also introduces some linear simplifications to the model. This leaves some uncertainties that can be quantified and included to improve realism of the model.

First of all, there is the uncertainty in the time data used, from which the time windows are extracted. The start and stop times are rounded to seconds in order to be usable within UPPAAL, since the model makes use of time steps of a second. It is also assumed that the given start and stop times are completely correct. These times are, however, based on the orbital flight trajectory of the satellite and there might therefore be an uncertainty in the accuracy of these times due to slight variations in the trajectory. The start and stop times might therefore be modeled as a time range with the highest probability at the given start or stop time.

Next, it is assumed that the connection between the satellite and the ground stations is working optimally during the entire transmission period. This is not very likely, so the transmission contains uncertainty that will result in a variable transmission time and therefore load on the battery. This could be modeled with a worst and best case transmission load and their respective probabilities.

In general it is assumed that the loads of various applications are constant over the full time period. For example, the background load is assumed constant during the whole time frame of the model. The preheating and slewing periods and loads are also considered as constant and the worst case scenario is taken for these values. The uncertainty in these values can be modeled in the same manner as the transmission load, by taking their respective worst and best cases and their associated values.

Uncertainties are also present with the insolation periods and loads. It is assumed, just like the discharging loads, that the charging load generated by insolation is constant during the entire time the satellite is subjected to the sun's radiation. This is however not completely realistic, since slight changes in attitude, radiation levels and the functioning of the solar panels are present. It is also assumed that the load is immediately changed when the attitude of the satellite changes, without taking a gradient into account. This also holds for the transition between the insolation and eclipse states. The uncertainty presented by the gradient can however not directly be modeled with probabilities, since it is a problem that has to do with the limitations of a state based model. The uncertainty of the insolation load can be modeled again as a worst case/best case probability choice.

A large set of uncertainties are ignored by the assumption that the battery behaves linearly. By introducing the schedule validation step, in which the schedule is tested with a non-stochastic KiBaM, part of the battery uncertainties are validated. The validation step can however be improved by using a custom stochastic KiBaM model as presented in [1].

There are some assumptions made when it comes to execution priorities. L- and X-band communications are executed in a certain ratio to make sure the amount of accumulated data does not exceed a certain limit. Communication is also not allowed when a window has already started, since the communication is deemed unusable when no full sets of data can be transmitted. When a data variable is introduced, the communication moments will be managed by the model based on data gathering and will result in a more realistic communication. The data gathering will probably contain uncertainty.

The proposed changes based on probability will create a more complicated model for both the scheduler as well as the battery. An more optimal schedule can however be obtained as the model will no longer only take the worst case scenarios into account. This will most likely relief the load on the battery, which will make the schedule validation step easier to pass.

VI. Probabilistic Behaviour in UPPAAL STRATEGO

In order to model a Priced Timed Automata, UPPAAL CORA can be used to obtain a proper model that can deliver real schedules and an optimal path to this schedule, using given cost conditions. UPPAAL CORA, however, does not take uncertainties into consideration and worst case conditions are needed to generate a usable schedule. It is on the other hand not always realistic to consider the worst case scenario, especially when efficiency is an important factor. To deal with such uncertainties, UPPAAL STRATEGO can be used. UPPAAL STRATEGO gives an environment where Stochastic Priced Timed Automata can be modeled, which means multiple scenarios can be modeled with their probability taken into account. This way, uncertainties within the model can be introduced. Based on given constraints, UPPAAL STRATEGO is able to generate different strategies to achieve the given goals. When given enough boundaries for the simulation, the optimal strategy can be determined from the given options by UPPAAL STRATEGO [2].

The biggest disadvantage of UPPAAL STRATEGO is the fact that it does not generate a concrete schedule. Since the goal of this assignment is to generate a schedule that can be used to plan real life transmissions with a satellite, UPPAAL STRATEGO might not be the best tool to do so. Since UPPAAL STRATEGO deals with stochastic systems, a time trace like generated by UPPAAL CORA can not be created. The computed strategies are shown in a cost related visualization, but the actual strategies can not be deduced from this.

Another limitation of UPPAAL STRATEGO is related to the modelling language that is used. It is not possible to use handshake synchronization due to the statistical nature of the model checker. This means a large amount of extra global variables and guards is needed to create a model that consists of multiple automata like our model. Furthermore, discrete jumps in the cost variable are not allowed in UPPAAL STRATEGO on transitions. This is neither used in our UPPAAL CORA model however, since the the costs are accumulated in the

penalty states.
The big advantage of using UPPAAL STRATEGO over UPPAAL CORA would be the implementation of uncertainties in the environment. This creates a larger set of potential strategies based on best and worst case scenarios. The found solutions in UPPAAL STRATEGO will therefore be more adaptive. It means a more energy efficient solution can be found using UPPAAL STRATEGO, since only using worst case scenarios will result in a non-optimal usage of the available resources. [3]

VII. Discussion

The largest trouble we had was dealing with the constraints of UPPAAL, since not a lot of it is well documented (or at least not very accessible). The largest being integer ranges and uncertainties with how to pass around and store data to use with functions. There were multiple times where we had to adjust our model because of previous assumptions of how things worked were wrong, or small details in the difference of the provided model to what we had to deal with (power and timing concerns with slewing). Debugging the model after we thought it was ready was very time consuming. As UPPAAL has no good error reporting, it took a long time to determine where some errors were located, and even then, many remain.

We had an issue with converting the traced schedule into suitable data for the KiBaM, as we had no tools available to do so without writing them ourselves, which did not have time to do. This also holds for the generation of a schedule in the CSV format, since a transformation from the timed trace to a properly readable file would be very time consuming without additional tools.

References

[1] M. Bisgaard, D. Gerhardt, H. Hermanns, J. Krčál, G. Nies, and M. Stenger, "Battery-aware scheduling in low orbit: The gomx–3 case," in *FM 2016: Formal Methods* (J. Fitzgerald, C. Heitmeyer, S. Gnesi, and A. Philippou, eds.), (Cham), pp. 559–576, Springer International Publishing, 2016.

[2] A. David, P. G. Jensen, K. G. Larsen, M. Mikučionis, and J. H. Taankvist, "Uppaal stratego," in *Tools and Algorithms for the Construction and Analysis of Systems* (C. Baier and C. Tinelli, eds.), (Berlin, Heidelberg), pp. 206–211, Springer Berlin Heidelberg, 2015.

[3] W. Ahmad and J. van de Pol, "Synthesizing energy-optimal controllers for multiprocessor dataflow applications with uppaal stratego," in *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques* (T. Margaria and B. Steffen, eds.), Lecture Notes in Computer Science, (Germany), pp. 94–113, Springer Verlag, 10 2016.