1. The approach is introduced in general terms, and the research questions are formulated correspondingly, however the research contributions are very much related to the Automotive domain.

**Response:** The actual contributions of the thesis are generic. These includes the contribution on improving requirements specification, formal analysis of Simulink models and the software-to-hardware mapping. In the latter case, it targets on AUTOSAR but it is stated in the paper that the approach can be extended to any component-based software development. Indeed, the use cases are from automotive. So, to conform to the generic description of the research goals, the thesis contributions are rewritten, however, the validation is still conducted on the automotive use cases. Please check out **Pg.35-39.**

1. The contributions related to requirements handling seem to be disconnected from lower level design and verification tasks. In this respect, the design + verification flow needs to be described. For example, explain that the design is hand-made, clarify that the synthesis of architectural requirements is not automated.

**Response**:

The connection between the specifications and the design is clarified better in the **Introduction** section, and **Contribution** sections, which also clarifies how the design artifacts come into play in the contributions.

1. The papers included in the collection are presented and illustrated with some details before the research contributions are explained. This creates some confusion, it could be worth moving the contributions at the beginning and then list the papers with a mapping to the corresponding contributions.

**Response:** addressed in Section 4.6

1. The research contributions’ figure contains graphic symbols that should be better clarified. Notably, the meaning of the solid arrows from Requirement Specifications to Software Allocation and Model Checking should be better explained.

**Response:** addressed in Section 4, Figure 4.1

1. Figure 4 is worth a more extensive discussion. Notably, the distribution of requirements encoding into corresponding boilerplates concepts does not give any additional information to the average reader. Please explain better the potential uses of such knowledge.

**Response:** The figure is replaced by a workflow figure, i.e., Figure 4.1. Now, it basically illustrates how the contributions connected to each other to realize an embedded software development that is assured through formal techniques and software-to hardware allocation via optimization methods.

1. The proposal should characterize better the 10% of the requirements that cannot be specified with ReSA. In particular, it is important for the reader to understand if ReSA cannot be used for the specification of important requirements, or if the same requirements might be reformulated such that they could be expressed with ReSA.

**Response:** addressed in Conclusion section. Basically, it is the limitation of the the language’s expressiveness, i.e., it cannot express specifications with quantification and timed operators. So, Peer-review should come into play to analyze the impact in doing so.

1. Please clarify the connections between ReSA editor and auto-completion features based on available modelling languages (e.g. EAST-ADL, internal UML profiles).

**Response**: I stated briefly in a single paragraph about this case, and then referred the reader to check Paper A, where it discusses the idea behind specialization of a system model to EAST-ADL model through subtyping (or inheritance) properties. Note, that the auto-completion is a result of the “specialization”, basically mapping the ReSA and EAST-ADL types. Please check Section 4.

1. When you discuss about allocation analysis algorithms, it is clear that over a certain limit the exact solution does not work anymore due to space explosion issues, however it is not completely understandable what concretely such a limit would be in terms of model sizes. Therefore, it would we worth clarifying when to use exact analysis and when to use heuristics.

**Response**: this is addressed in detail in Sec 4.4, and 4.5 contributions. The limitation of the exact approach as discussed in Sec 4.4 quantifies the software in terms of components, chains and tasks, and shows the bottleneck. This issue is addressed in Sec 4.5, which also shows the results.

1. Some minor comments have been annotated directly on paper and passed to Nesredin.

**Response**: addressed