The development of highly distributed Systems of Systems (SoS) poses a big challenge on the whole development process of such systems. Especially in Requirements Engineering, one has to cope with the resulting variety of stakeholders and their multitude of different and possibly contradictory goals. This is challenging for requirements elicitation, documentation, and management, especially with regard to communication and consistency.

 The development of CPS requires the collaboration of different disciplines, like mechanical engineering, electrical engineering and computer science. This cre- ates new challenges for Requirements Engineering (RE), which needs to estab- lish a common perception of the targeted CPS for the involved stakeholders. This paper will elaborate the specific challenges in RE for CPS based on a lit- erature review. Natural Language Processing (NLP) is used as an approach to automatically translate shared informal requirements specifications to formal domain specific models for the involved disciplines, to develop a comprehen- sive RE methodology for CPS.

Systems have to be both appropriate and cost effective [6], which makes understanding the requirements of the customer and other affected stakeholders a prerequisite for successful systems engineering [7]. They are needed for planning the development process, assessing the impact of changes and testing the acceptance of the outcomes [8]. Inadequate Requirements Engineering (RE) is one of the main sources for the failure of development projects and culminates in exceeding budgets, missing functionalities or even the abortion of the project [9]. Consequently, in concordance with the principles of concurrent engineering, RE continues along the development process of a system and secures a consistent and traceable elicitation and management of requirements. There is an ongoing interaction between RE and the development phases in systems engineering

Therefore, adequate RE is also the key to success or failure of every CPS development project. However, CPS differ from conventional production systems in various aspects, leading to new challenges for the RE process. CPS are open systems, which have to be aligned with dynamic user needs in a global context. Furthermore, requirements towards CPS underlie evolutionary changes. The scope and emphasis of the relevant requirements change with respect to the final application and environment of the CPS [4]. Finally, CPS are based on integrating hardware, software, and service components, covering the whole life cycle, from ideation to decommission. The required competencies for CPS development and their support in all life cycle phases have to be included through collaboration with partners from the different disciplines

In contrast to embedded systems where the focus is more on the computational elements, CPS emphasize the link between the computational and physical elements. In this sense, CPS represent a network of interacting elements with physical input and output instead of as standalone devices

After explanation of V model Requirements are important for all layers in systems engineering. It is necessary to validate requirements from lower layers against requirements from upper layers and the stakeholder needs in order to check that the requirements represent the original goals for the system development. Furthermore, the design and implementation of the system has to be verified to check that it fulfills the requirements. In order to support the different tasks, the specification of requirements has to follow several contradictory objectives. To minimize the time to write requirements and make them understandable for all of the involved stakeholders, often an informal approach is used without any constraints on how requirements are specified e.g. in natural language. However, to minimize the time needed to validate requirements and verify the system design, a formal specification is required. Abstract semantics and syntax enable automatic checks like formal verification. Often a trade-off between formal and informal specification is implemented, e.g. by using controlled languages [18] or boilerplate techniques [19]

three distinct aspects of a requirement:

1)Who wants the functionality;

2) What functionality the end users or stakeholders want

the system to provide; and

3) Why the end users and stakeholders need this functionality

(optional).