### Can poor models give good controllers?

An important  application of model approximation amounts to deriving simplified models for control system design. Indeed, if simulation or first-principle models become too complex to allow model-based controller synthesis, then a simplified substitute model is often helpful to enable controller synthesis. This project aims to derive novel model approximation techniques that take the performance objectives and specifications of the controlled system into account in an explicit way. This means that we aim to derive model reduction techniques that provide quantified performance and robustness guarantees on the controlled system that is synthesized on the basis of the reduced model.

1. Assessing the credibility of crowdsourced testing
2. https://www.wur.nl/en/Research-Results/Chair-groups/Social-Sciences/Information-Technology-Group/Thesis-Topics/MSc-Thesis-topics-to-be-updated.htm
3. [**Coupling Design Thinking, User Experience Design and Agile**](https://www.researchgate.net/publication/320623595_Coupling_Design_Thinking_User_Experience_Design_and_Agile)
4. [**Models for Innovative IoT Ecosystems**](https://www.researchgate.net/publication/320623589_Models_for_Innovative_IoT_Ecosystems)
5. [**A virtual reality based internet-of-things (IoT) framework for micro devices assembly**](https://www.researchgate.net/publication/320742745_A_virtual_reality_based_internet-of-things_IoT_framework_for_micro_devices_assembly)
6. Generation of Automatic Test Cases with Mutation Analysis and Hybrid Genetic Algorithm, **2017, IEEE**
7. Analyzing the Behavior of Software Architecture
8. Describing the Evolution of Software Architecture
9. Architectural Paradigms for the Internet of Things
10. Code Quality Management tools
11. DevOps
12. IoT
13. A Survey on Security and Privacy Issues in Internet-of-Things
14. A survey of search-based refactoring for software maintenance <https://link.springer.com/article/10.1186/s40411-018-0046-4>
15. Guest editorial foreword for the special issue on automated software testing: trends and evidence <https://link.springer.com/article/10.1186/s40411-018-0047-3>

2) Natural language processing (NLP) for managing software requirements. The problem is to study how natural language processing can be applied in requirements management in the case of existing requirements.  For example, how the quality of requirements can be improved, how the requirements can be refactored, and how dependencies or other relationships can be extracted. The study should cover literature but also include an empirical part using, e.g., Qt's Jira (see below) or some other, preferable opens source, requirements data. As a starting point, see e.g. [http://dl.acm.org/citation.cfm?id=2976769](http://dl.acm.org/citation.cfm?id=2976769%C2%A0)(Extracting domain models from natural-language requirements: approach and industrial evaluation, Arora et al. 2016).

3) Diagnosis algorithms for deficiencies in requirements or domain models. Domain models are high abstraction level models that consist of, e.g., features, requirements, or high level software components or services. Such a model can be constructed manually or automatically from smaller pieces. A feature model is an example of domain model that can be constructed from requirements automatically, or it can be constructed manually.  A domain model can suffer from different kinds of deficiencies, such as a lack of relationships or other information, or conflicting information. The special focus could be on inconsistent constraints. The objective is primarily to study and apply the existing algorithms to a domain model, and assess their practical value. The work should cover an empirical part, in which algorithms are applied to an open source system. See e.g., An efficient diagnosis algorithm for inconsistent constraint sets, Felfernig et al 2012 AI EDAM.

4) Ontologies in the systems used for requirements engineering . Nowadays, various different systems are used to manage requirements especially for large scale projects. For example, Qt open source project uses Jira issue tracker to manage requirements. That is, a requirement is reported as an issue, the issue is checked and finally assigned to a specific release. Respectively, Eclipse open source project uses Bugzilla. Requirement engineering ontologies or more practically datamodels or schemas describe what should be documented about requirements.  The research problem is that what kinds of data about requirements can or should be managed, and how data is managed in practice especially in an open source project. The focus can be adjusted, but the topic can consists of the following parts 1) A survey and synthesis of ontologies reported in literature.

2) Empirical comparison of ontologies in open source project, specifically Eclipse's Bugzilla and Qt's Jira. 3) Empirical investigation on how ontologies are actually used in the above cases. For example, Bugzilla includes a dependency field but how extensively (often) the field is actually used?

* Efficient and Rapid Machine Learning Algorithms for Big Data and Dynamic Varying Systems
* The plant disease detection using glcm and KNN classification in neural networks
* To propose enhancement in checkpoint algorithm for fault removal in DCS
* MATLAB spectral clustering package
* The secure and efficient data routing technique for internet of things

1. Assessing the credibility of crowd sourced testing
2. ***RESEARCH ISSUES IN IOT:*** ***https://www.linkedin.com/pulse/thesis-ideas-internet-things-advantages-applications-manpreet-kaur/***
3. *Device management*  
   *Sensor data acquisition in management*  
   *Industrial applications based on IoT*  
   *Integration of data from multiple sources*  
   *Device diversity and interoperability*  
   *Flexibility and evolution of applications*  
   *Scale, data volume, and performance*  
   *Scalability in networking, storage and computation to handle exponential growth of data volume from sensors*  
   *Security of the data-at-flight and data-at-rest without compromising on scalability*  
   *Preservation of privacy of the user data and properly balancing between privacy and utility*  
   *Interoperability among myriad sensor data sources (physical communication level, network level, data syntax level, and data semantics level)*  
   *Rich analytics and visualization (generic, sensor-specific, and domain-specific) provided in real-time, as required etc.*
4. IoT in modern production chains.
5. IoT the new concept that makes little things communicate
6. Using IoT to add value to healthcare services
7. Monitoring life with IoT
8. 1. Big IoT data analytics
9. 2. IoT based Smart agriculture cloud service
10. 3. IoT based tourism management
11. 4. IoT based geological structure analysis
12. 5. Connected society based on IoT
13. - Programmability challenges for resource-constrained IoT devices
14. - A proxy framework for trustworthy API interoperability in the IoT - SDN and NFV for the IoT in Resource-Constrained Devices
15. - Development and implementation of smart IoT gateways
16. - Development of lightweight and secure IoT device management framework

# The role of the Internet of Things in the improvement and expansion of business