2023 Conference on Systems Engineering Research

A Framework on Early Decoupling Level Metric Assessment based on NLP4RE

Lu Xiao, Gengwu Zhao, Maximilian Vierlboeck, Roshanak Nilchiani

School of Systems & Enterprises, Stevens Institute of Technology, Hoboken NJ

Abstract

Modularity is recognized as a general and effective solution to tackle the complexity of system of systems in modern society. Despite the value and benefits of modularity, it has been a known challenge to measure and evaluate the modularity of complex systems in real-life, especially early in the development life cycle such as in requirement engineering phase. This paper proposes a framework that enables early assessment of the modular structure of a system based on system requirements. The framework combines 1) a Natural Language Processing for Requirements Engineering (NLP4RE) application developed together with the Systems Engineering Research Center (SERC), 2) a Decoupling Level metric, and 3) a novel Ablation procedure to quantify the decoupling and coupling effects of each design element in a system. We conduct a case study on an unmanned aircraft system (UAS) from SERC, for which the third and fourth authors of this paper have in-depth knowledge about. From this case study, we yielded interesting findings that align with expert's intuitive understanding of the UAS. The case study indicates that our framework can provide early and quantitative assessment of the modular structure of a system based on the requirements. It has the potential to assist the architect in reasoning and designing a modular structure by reviewing the decoupling and coupling effects of system requirements and key terms.

Keywords: Nature Language Processing, Requirements Engineering, Modularity, Design Rule, Decoupling Level

1. Introduction

Modularity is recognized as a general and effective solution to tackle the complexity of system of systems in modern society. Carliss Y. Baldwin and Kim B. Clark[4] developed a powerful theory of design rule based on the computer industry as an example. According to their theory, any complicated systems can be designed, constructed, interpreted, and analyzed as design rules and modules. The design rules define the high-level design decisions that decouple the other design elements in a system into modules. This modularity in system design provides flexibility if high-level design rules are established and obeyed. Modularity offers numerous benefits, including but not limited to 1) a way of managing complexity—instead of dealing with a gigantic monolithic system, the complexity is broken down into smaller and manageable modules; 2) parallelization in development—development tasks can be divided based on modules and conquered with parallel efforts from sub-teams, which may even be geographically distributed; 3) promoting flexibility for technology upgrades and preventing vendor lock-in.

Despite the value and benefits of modularity, it has been a known challenge to measure and evaluate the modularity of complex systems in real-life.[5] To our best knowledge, there is no existing methods to conduct an early assessment of system modular structure based on system requirements. If a system architect can quantitatively assess the modular structure of a system based on the system requirements, this could shed light on how they design the system to best promote modularity and its benefits. For instance, if a particular requirement is found to have a strong coupling effect---meaning it connects many dependencies with other requirements, an architect could reason about how to split this requirement into effective decoupling design rule and respective modules following Carliss Y. Baldwin and Kim B. Clark's theory. This could potentially save a significant amount of future rework if the modularity assessment happens much later in the development process.

To fill this gap, this paper proposes a framework that enables early assessment of the modular structure of a system based on system requirements.[11] On the one hand, we leverage a Natural Language Processing for Requirements Engineering (NLP4RE[11,12]) application developed together with the Systems Engineering Research Center (SERC) [1]. This application identifies the dependencies among system requirements and its key terms based on natural language processing techniques. And we use a Design Structure Matric[6] to represent the dependencies among requirements. On the other hand, we leverage a metric called Decoupling Level (DL)[3,14] to measure to what extent the system requirements and key concepts are decoupled into small and manageable modules. Furthermore, we contributed a novel procedure, called Ablation DL, to quantitatively evaluate and rank the contributions of different design elements to the modular structure of an entire system. The overall design rationale of this procedure is that, by removing an element from a Design Structure Matrix at a time, the change of the DL metric resulting from this removal quantifies the contribution of the element to the overall modular structure of the entire system.

To evaluate the effectiveness of this framework, we conducted a case study on an unmanned aircraft system (UAS) from SERC[13], for which the third and fourth authors of this paper have in-depth knowledge and expertise about. From this case study, we yielded interesting findings that align with the system expert's intuitive understanding of the UAS.

First, the overall DL metric of the UAS is 0.4395 (See details in Section 2.2), comparing the "health chart" composed of the DL metrics of 129 software systems provided by Ran et. Al[3], the modular structure of our case study subject UAS is below average. By applying the Ablation_DL on both DSM_req and DSM_term, we noticed that most of the requirements (87%) or the design terms (71%) do not have an obvious impact on the DL metric. However, there are a few requirements or design terms that contribute to decouple or couple the modular structure of the system. These requirements or design terms should be of high interest for a system architect if he/she is looking for opportunities to improve the modular structure. Based on our in depth understanding of the case study subject, we can reason how and why some of the requirements or design terms have a decoupling or a coupling effect to the entire modular structure of the system. The case study indicates that our framework can provide early and quantitative assessment of the modular structure of a system based on the requirements. It has the potential to assist the architect in reasoning and designing a modular structure by reviewing the decoupling and coupling effects of system requirements and key terms.

2. Approach Background

2.1. Natural Language Processing for Requirements Engineering (NLP4RE)

The approach that enables the application of the DL metric in the paper at hand is based on a Natural Language Processing for Requirements Engineering (NLP4RE) application developed together with the Systems Engineering Research Center (SERC). This approach is presented in detail by Vierlboeck, Nilchiani and Blackburn[1,9,10] and summarized hereinafter.

The used approach combines a unique application of Natural Language Processing for the identification and organization of all entities within a requirement specification that is available in text form. By applying the NLP process presented in the Figure below, the two types of structure can be deduced from a set of requirements (in addition to a hierarchy structure, if available): 1) the structure based on the terms and entities in the requirement text and 2) the connections of the requirements based on the entities.

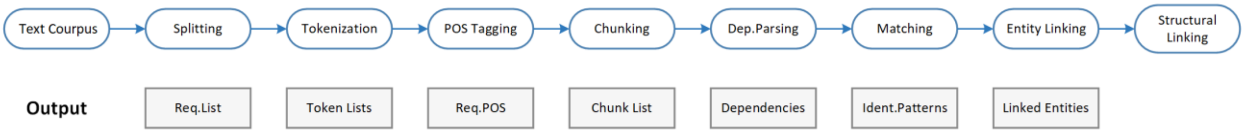

Fig. 1. - NLP Flowchart and Process

The result of the process above is a network of terms/entities that are connected based on their sentence structure and text relationships. In addition, since the different entities are part of individual requirements, these requirements can also be linked based on their content.

Since the representation of the results of the NLP approach is a network and thus underlying structure, the results can be also represented by a symmetrical adjacency matrix (inclusion of directional information is possible, but currently considered work in progress). This adjacency matrix resembles the structure of a DSM and based on the requirement specification that serves as a source, it is a representation that can be used as an input for the DL approach. Thus, the network and respective adjacency matrix is directly fed into the DL application on both levels, the requirement level and the term level as two separate matrices DSM_req and DSM_ term respectively.

2.2. Decoupling Level Metrics

Ran et. al. proposed a metric called Decoupling Level (DL) [3], based on the concept of design rules and modules defined by Carliss Y. Baldwin and Kim B. Clark[4]. The DL metric measures how well a system is decomposed into small and manageable modules. The DL metric is built upon the notion of Design Structure Matrix (DSM) of a system also proposed in the book of Carliss Y. Baldwin and Kim B. Clark[4]. Any complex system can be represented as a n*n square matrix, called DSM. The elements in a DSM represent the design elements in a system; while the cells represent the dependencies among the elements. For example, Figure 2 is such an example, showing the DSM of a software program for math calculators. The rows and columns, arranged in the same order, represent the design elements—the java source files. The cells represent the code reference among the source files. For instance, cell [2, 1] indicates that Answer_java depends on UI_java. As we can see from this example, the high-level design rules are arranged on the top four rows, which decouple the remaining elements into 6 mutually independent modules in the bottom layer that contains elements from row 5 to row 16. In the DL calculation, a module with five or less elements is considered a true module, referencing the cognitive theory. Overall, DL measures the percentage of design elements in a system that can evolve as "true" modules. The higher the DL metric of a system, it means that the system is better decoupled into small and manageable modules that handle complexity, support parallelization, and promote flexibility.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	UI_java	(1)															
2	Answer_java	x	(2)	x													
3	Question_java	x	x	(3)													
4	Survey_java			x	(4)												
5	SaveLoadFile_java				x	(5)											
6	TextFileUI_java	x					(6)										
7	CommandLineUI_ja	x						(7)									
8	Letters_java								(8)								
9	Match_java	x	x	x					x	(9)	x						
10	MatchingAnswer_ja	x	x	x					x	x	(10)						
11	ChoiceAnswer_java	x	x	x								(11)	x				
12	Choice_java	x	x	x								x	(12)				
13	EssayAnswer_java	x	x	x										(13)	x		
14	Written_java	x	x	x										x	(14)		
15	Test_java		x	x	x											(15)	
16	AnswerSheet_java															x	(16)

Fig. 2 An example of showing the DSM of a software program for math calculators

Ran et. al[3] has shown in their study of hundreds of real-life software projects that the DL metric is capable of supporting 1) cross-project comparison of subjects of different characteristics; 2) monitoring the evolution of software systems by reflecting major restructuring activities performed by developers; and 3) reflecting real maintainability of software projects, which is independent from the calculation of DL metric of a system. In addition, Xiao et. al[2] also extended the adoption of the DL metric beyond purely software systems. They calculated the DL metrics of two open-source systems in the context of cyber physical systems, namely OpenWrt and MD PnP. They observed that the DL values of these two projects are higher than about 85% and 70% of 129 software projects from Ran et. al.[3]'s study. The implication is that both OpenWrt and MD PnP are better decoupled into mutually independent modules when compared to the majority of the previously studied pure software projects. The conjecture was that hardware components in a cyber-physical system increase the overall complexity of the system; increasing the modularity of the software may be a way to cope with the complexity.[7]

In this study, we are further extending the adoption of DL metrics to go beyond software systems. We apply the DL calculator built by Ran et. al.[3] on the DSM constructed from the system requirements of a landing gear system. This system stems from a model-based systems engineering (MBSE) research project of the Systems Engineering Research Center (SERC) and includes an unmanned aircraft system (UAS). From this UAS, the requirements were used as per the process in Figure 3. In the process, the two output DSMs (one for the term and one for the requirement level) were analyzed in the DL approach. As we will describe later, we are proposing a new algorithm built upon the DL metric to help us compare and rank the relative contributions of different design elements in the system requirements to the modular structure of a system.

3. Our Framework

With the combination of the two techniques introduced above, an iterative and recursive process is enabled that is shown in the figure below.

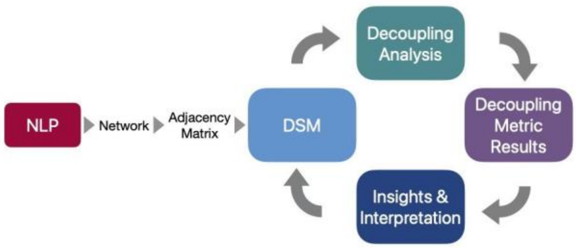

Fig. 3. - Recursive and Iterative Circular Process

As seen in the figure, the input from the NLP approach is used in the form of an adjacency matrix as DSM. This matrix is then analyzed with the decoupling method (elaborated in detail in Section 3.1) and the results interpreted (discussed in Section 4 and Section 5). With these results then, improvements and potential changes are possible and can be recommended, which then can be analyzed again regarding their impact. This process can be repeated as needed and whenever needed, so the improvements can be strategically placed and used for optimization throughout the development lifecycle.

The crucial contribution of this paper, namely the connection from the NLP to the Decoupling Analysis through the Adjacency Matdix/DSM is made possible by the fact that the connections elicited through the NLP represent the apppearances of different enitites and or elements within the requirements. Thus, we argue that if an entity is identified/found in separate instances, it is part of the connections resulting from all respective instances. While this approach cannot claim to capture all connections that exist since implicit connections are possible as well, the captured information is necessarily a significant subset of all connections within the system and thus can be used as a foundation for the presented analysis. The analysis can later also be expanded. Furthermore, the validity of the results shown in Section 5 is not reduced by the fact that not all connections might be captured just yet since the produced insights are still valid and still the minimum decoupling facts to be considered. Yet, the limitations also described in Section 5 still must be considered. Currently, scalability is being worked on since larger systems pose bigger challenges for the elicitation of an Adjacency Matrix/DMS from the NLP algorithm.

In the next subsection, we elaborate the details of the “Decoupling Analysis” component in the framework

3.1. Decoupling Analysis

In this study, we are proposing a novel algorithm that combines an ablation procedure and the DL calculation to quantitatively evaluate and rank the contributions of different design elements to the modular structure of an entire system. The overall design rationale of this algorithm is that, by removing an element from a DSM at a time, the change of the DL metric resulting from this removal quantifies the contribution of the element to the overall modular structure of the entire system. For example, if by removing element “x” from a DSM, the DL metric increases from 0.5 to 0.6, it indicates that “x” contributes a 0.1 decrease in the DL. Put differently, “x” contributes negatively to the modular structure of the system, and it serves as a “coupler” that tangles modules in the system. In comparison, if the DL metric decreases after the removal of “x” from the DSM, it indicates that “x” contributes positively to the modular structure of the system. In other words, “x” serves as a decoupler to the system. This quantitative measure with the ablation process enables the ranking of all the design elements in a system. A system architect or designer can reflect based on the quantification, regarding where and how to improve the modular design of a system. For example, to most effectively decouple a system into modules, one should examine the design element that has the largest negative impact on the DL metric of the system. By restructuring or breaking down this design element, one could possibly expect to maximize the decoupling of the system into modules. Such broken-up design elements would then appear as different entities and increase the entity and connection count, which in turn increases the system's decoupling.

Table 1. Pseudocode for the algorithm Ablation_DL.

Algorithm Name	Ablation_DL
Purpose	Evaluate and rank the contribution of design elements in a dsm to the modular structure of the entire system through an ablation procedure.
Input	**dsm:** The dsm matrix to be analyzed
Method	Get_DL_Value(dsm): return the **DL value** of the **dsm** matrix get_dsm_elements(dsm): Returns a **list** containing all the elements of the dsm matrix remove_dsm_element(dsm, element): Removes the **element** from the **dsm** matrix and returns the new dsm matrix without the removed element.

Process	Origin_DL=Get_DL_Value(dsm)
	elements=Create_table(column=[‘element ’,’delta’])
	
	Raw_elements_list= get_dsm_elements(dsm)
	For each **element** in Raw_elements_list:
	    new_dsm_by_remove=remove_dsm_element(dsm, element)
	    new_DL=Get_DL_Value(new_dsm_by_remove)
	    delta=Origin_DL-new_DL
	elements.addRow(column[‘‘element’]=**element** , column[‘‘delta’’]=**delta**)
	
	ranked_elements=elements.sort(by=column[‘‘delta’’])

Output	**ranked_elements**:The table sorted by value. key is each **element** in the dsm matrix input. value is **delta**, the change of the DL value.

The Table 1 illustrates the details of the ablation procedure based on the DL metric. We name this procedure Ablation_DL. It takes as input the DSM of a system. The output is a ranked list of all the design elements in the system, together with the quantification of the impact of each element on the DL metric of the system. Ablation_DL is built upon several existing procedures, including:

- DL_Calculator(dsm), which calculates the DL metric of a given dsm
- get_Elements(dsm), which return all the design elements in a dsm

- remove_Element(dsm, element), which creates a new dsm by removing the given design element from the input dsm.

The Ablation_DL first calculates the original DL metric of the input dsm, namely Origin_DL using the DL_Calculator. Next, it iterates through each element in the input dsm from get_Elements(dsm). In each iteration, it generates a new dsm, namely new_dsm_by_remove, by removing the current element from the original dsm, i.e. remove_Element(dsm, element). A new DL metric is calculated for the new_dsm_by_remove. The delta, i.e. difference between the Origin_DL and new_DL is recorded to quantify the impact of the current design element to the modular structure of the entire system.

- If delta < 0, it indicates that the design element is a coupler, since it has a negative impact on the modular structure of the system and removing it will increase the decoupling level.
- To the contrary, if delta > 0, it indicates that the design element is a decoupler, since it has a positive impact on the modular structure of the system and removing it will decrease the decoupling level.

Finally, Ablation_DL sorts the design elements in the input dsm based on the delta_DL of each element. This provides a list of prioritized design elements for an system architect to review and inspect the system design for better modular structure.

4. Case Study Subject and Key Findings:

We conducted a case study by applying the Ablation_DL on a UAS system from the SERC. We used the NLP approach described in the background section as a foundation and to first construct two DSM(s) to represent the design structure of the system requirements for the UAS system. One of the DSM is at the requirement level. There are a total of 70 requirements, extracted from the requirement specification of the UAS. The DSM captures the semantic dependencies among the 70 requirements in the system. We denote this DSM as DSM_req, which captures the coarse design structure of the UAS system reflected in the high-level system requirements.

Furthermore, the 70 requirements are further decomposed into 245 key terms through NLP (Nature Language Processing) techniques. These 245 terms capture the essential design concepts embedded in the high-level requirements. We also constructed a DSM that captures the semantic dependencies among these 245 terms of the UAS system. We denote this term-level DSM as DSM_term, which captures the fine-grained semantic design structure of the key concepts in the system requirements.

We first calculate the overall DL metrics of both DSMs, namely DSM_req and DSM_term. The overall DL metrics provide some quantitative impression of the current modular structure of the system based on system requirements. Next, we apply the Ablation_DL on both DSMs of the system. This helps us understand how the different requirements and design elements contribute to the decoupling level of the modular structure of the system. Following we summarize the key findings from our case study. We envision that a system architect could leverage the DL metric and the Ablation_DL procedure to draw quantitative measures and get insights about the design of their system. The analysis could be conducted as early as system requirement analysis. This shed light on the system's modular structure even before putting forth a significant amount of implementation effort.

- Finding 1: The overall DL metric of the UAS is 0.4395 at the requirement level and 0.4193 at the design term level. Comparing the “health chart” composed of the DL metrics of 129 software systems provided by Ran et. al[3], the modular structure of our case study subject UAS is below average. This provides an overall quantitative impression of the modular structure of the system in hand. Overall, the results are as expected given the fact that the case study at hand addresses a highly integrated physical system. The structure and architecture of the system is by design not focused on modularity and this focus is represented in the DL metrics as a result.
- Finding 2: By applying the Ablation_DL on both DSM_req and DSM_term, we noticed that the majority of the requirements (87%) or the design terms (71%) do not have an obvious impact on the DL metric. This indicates that a system architect could possibly give lower priority to these requirements or design elements due to their relative low importance to the design structure.
- Finding 3: There are a few requirements or design terms that contribute to decouple the or couple the modular structure of the system. These requirements or design terms should be of high interest for a system architect if he/she is looking for opportunities to improve the modular structure.

- Finding 4: Based on our in depth understanding of the case study subject, we can reason how and why some of the requirements or design terms have a decoupling or a coupling effect to the entire modular structure of the system. For example, Term “landing gear” has the lowest delta, which is -0.1381. It means that after removing this term, the DL value of the UAS increases from 0.4193 to 0.5574. Thus, it has a coupling effect on the modular structure of the system. Similar to Finding 1, this is to be expected and further validates the approach since the landing gear poses the central part of the assembly and as such must have a coupling effect. A removal of this part would fragment the entire network and assembly respectively, which would yield more individual modules and thus a higher DL level. In general, a system architect should be recommended to review this design element first if the goal is to decouple the system.

5. Case Study Results:

In this section, we present the findings of our case study with more details following four key RQs:

5.1. Research Questions:

- RQ1: How is the DL metric of the UAS system compared to the DL metrics of 129 software systems provided by Ran et. al[3] This RQ helps us to quantify the overall modular structure of the UAS based on its requirements. By comparing to the “health chart” formed by 129 software projects, we can have a quantitative impression of how modularized the system is based on its requirements.
- RQ2: How do the different requirements in the UAS system contribute to the DL metric? Can we identify decoupler and coupler requirements of the system? This RQ helps us to reveal the requirements that have the most positive and most negative impact on the modular structure of the system. If the Ablation_DL procedure finds all the requirements have the same contribution, it does not help an architect to prioritize requirements for the goal of improving the modular structure.
- RQ3: How do the different design terms in the UAS system contribute to the DL metric? Can we identify decoupler and coupler terms of the system? This RQ provides similar information as RQ2 but in finer granular details. An architect could prioritize the design concepts that contribute most negatively to the DL metric to improve the modular structure of a system. Again, if the Ablation_DL procedure finds all the terms have the same contribution, it does not provide much insight for an architect.
- RQ4: Can we reason why some requirements and design terms have a decoupling effect and some have a coupling effect to the UAS system based on our understanding of the system? This RQ aims to verify if the top prioritized requirements or terms using the Anbaltion_DL matches with our in-depth understanding of the UAS. If so, it implies that the proposed Anbaltion_DL process should be able to provide valid insights for ranking the contribution of the design elements to the modular structure of a system.

5.2. RQ1:

Figure 4 (a) and Figure 4 (b) show how the DL of the DSM_req and of the DSM_term compared to the “health chart” of 129 software systems from prior study[3,15]. As we can see, the decoupling level of this UAS is at 20 percentiles, meaning it is more coupled than 80% of software projects studied prior. This is not surprising since the UAS is a highly physical system, and the parts are by their nature more coupled. With our Ablation_DL process, removing the most coupling requirement (requirement #53) of the system could improve the DL by more than 5%, which is significant compared to where the project is. We observe quite consistent observations with the DSM_term, as illustrated in Figure 4(b).

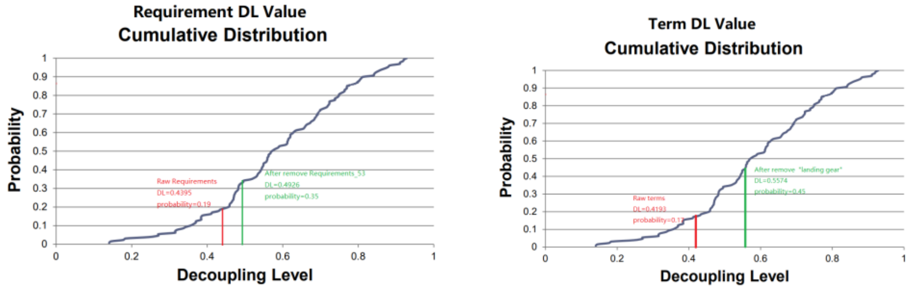

Fig. 4. (a) Raw DL value and improved DL value of requirement (b) Raw DL value and improved DL value of Term

5.3. RQ2:

Figure 5 illustrates the distribution of the delta DL by removing each and every requirement from the DSM_req. As we can see, the majority of the requirements have quite a trivial impact (delta within [-0.002, 0.005]) on the DL metric of the system. This indicates that these requirements do not have an obvious impact on the overall modular structure of the UAS. In comparison, there are a few requirements that manifest a more significant impact on the DL metrics of the system. In particular, 6 of the 70 requirements have an obvious positive impact on the DL metric of the system, since their delta is negative (< -0.02). In comparison, 5 of the 70 requirements have an obvious negative impact on the DL metrics of the system, since their delta is positive (> 0.05). More specifically, requirement #53 and requirement #7 has the lowest delta, which is -0.053. It means that by removing the requirement from the system, the DL value increases from 0.4395 to 0.4926. In comparison, requirement #25 has the highest delta, which is 0.0226. It means that after removing this requirement, the DL value declines from 0.4395 to 0.4169. We will provide a more detailed discussion of what are these requirements, and the reasoning behind their impact on DL based on our understanding of the UAS in RQ4.

5.4. RQ3:

Figure 3 illustrates the distribution of the delta DL by removing each and every design term from the DSM_term.

We yield to similar observations as for the DSM_req. That is, the majority (174 out of the 245) of the terms only have quite trivial impact on the system DL. For inspecting the modular structure and identifying opportunities to improve the modular structure, a system architect should prioritize the few terms that have more significant negative contributions to the DL metric. For example, Term "landing gear" has the lowest delta, which is -0.1381. It means that after removing this term, the DL value of the UAS increases from 0.4193 to 0.5574. Interestingly, the term "landing energy absorption" has the highest delta, which is 0.0147. It means that after removing this term, the DL value declines from 0.4193 to 0.4046.

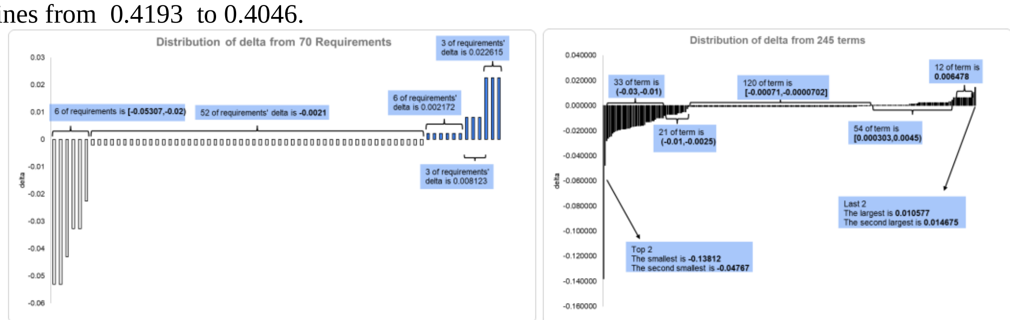

Fig. 5. (a) delta distribution of requirement (b) delta distribution of term

5.5. RQ4:

We reviewed the top 3 most coupling requirements and top 3 most decoupling requirements in the DSM_req. As discussed earlier, a requirement is considered decoupling if removing it from the system decreases the decoupling level of the system; while a requirement is considered coupling if removing it from the system increases the decoupling level of the system. The goal here is to reason why some of the requirements are decoupling and some are coupling, and if the identified items match with our in-depth understanding of the UAS.

When looking at the requirements behind the IDs, we see for the coupling ones that they are long and descriptive which increases the number of connections that they introduce. Also, the terms included within the requirements are highly connected themselves, meaning that they connect to many other requirements. Thus, it is valid that these requirements couple the system more since they bring fragmented parts together and thus integrate the system more. Such requirements, while potentially necessary and unavoidable when it comes to their effect, can be crucial and hard to verify. Thus, depending on the type of system in question, such high degrees of coupling might merit reevaluation.[8]

As for the top three decoupling requirements, they are shorter in nature, with 61 and 61 only containing two entities. Moreover, the terms within the requirements are very specific and form the end of branches, which decouples the system. While the opposite is true for these requirements regarding verification, too many branch ends can indicate a very broad and distributed architecture, which might not be ideal in some cases and thus the DL metric can serve as a second check to align the structure with the needs, goals, and objectives as well.

Table 2. Requirement level and Term level

Requirement level				Term level			
Top 3 Coupling Requirement		Top 3 Decoupling Requirement		Top 3 Coupling Term		Top 3 Decoupling Term	
Requirement ID	Delta	Requirement ID	Delta	term name	Delta	term name	Delta
# 53	-0.0531	# 61	0.02262	landing gear	-0.13812	landing energy absorption	0.014675
# 7	-0.0531	# 62	0.02262	time	-0.04767	landing gear actuation	0.010577
# 52	-0.0429	# 25	0.02262	inches	-0.02815	single point non-structural failure	0.009714

Similarly, we conducted the same inspection of the most coupling and decoupling design terms in the DSM_term. Overall, we find the top ranked terms align with our intuitive understanding of the UAS.

Similar to the analysis on the level of the requirements, the assessment of the terms individually produced an expected picture, which confirms and validates the approach. The three top coupling terms are high in connectivity, with the term “landing gear” being the central hub of the architecture that everything stems from. Thus, this coupling effect is expected and necessary given the system at hand. As for the two other terms, their popularity and thus coupling effect is also explainable since they refer to universally used measurement units that inevitably appear in more than one requirement. This shows a limitation of the NLP approach at this time, which is that more rules are necessary to differentiate between terms on a semantic level instead of a purely linguistic one.

Regarding the top three decoupling terms, we see that they all are long compound terms that stand on their own and thus have one connection upstream and maybe a second one downstream. This low degree of connectivity makes them very decoupling, which is also valid. Yet, as the second term shows, a cross-connection to the term “landing gear” might be logical depending on the system in question since the term is contained in the compound noun. Yet, such distinctions cannot be made universally and are thus still subject to ongoing research.

Also, it has to be noted that general terms, such as ‘time’ and ‘inches’, which appear in more than one requirement, are not necessarily connected and should be assessed after the NLP analysis. Due to their general nature, appearance and thus connections are likely and thus, their coupling level is to be expected to some extent. One possible solution to prevent such false positives would be to exclude general terms from the coupling side, which is currently being researched. Also, the problem of false positive connections increases with a larger scale of the system since the appearance of similar terms becomes more likely in a larger system even when two entities are not the same and thus should not be treated as the same part/design element.

All in all, the results of the requirement and term level assessment are as expected given the system at hand and thus validate the DL metric and approach. While certain limitations might still be visible, they are related to the input and can thus be addressed without compromising the validity of the approach showcased in this paper.

In addition, the insights generated by the application of the presented approach greatly increase the understanding of the system and multiple subsequent analyses are possible. For instance, the top terms identified could be cross-checked with the expectations of experts and can also inform change management as well as design decision since a highly coupled or de-coupled system needs to be considered with these characteristics. Overall, this also allows for follow-up research and further contributions that are currently being worked on.

6. Conclusion:

In this paper, we contributed a framework that enables early assessment of the modular structure of a system based on NLP4RE technique. The framework is a combination of an NLP4RE technique, a Decoupling Level Metric, and a new Ablation_DL procedure. A case study on a UAS indicates that our framework has the potential to provide critical and quantitative analysis of the decoupling and coupling effects of different requirements and design concepts in a system. The quantitative analysis of our framework aligns with expert's knowledge of the case study subject. This indicates that our framework can provide help to system architect in best designing the modular structure of the system based on the analysis of system requirements.

Acknowledgement

This work was supported in part by award CCF-1823074 from the National Science Foundation.

References

1. Vierlboeck, M., Nilchiani, R., & Blackburn, M. (2022). Natural Language Processing to Assess Structure and Complexity of System Requirements.
2. Xiao, L., Pennock, M. J., Cardoso, J. L., & Wang, X. (2020). A case study on modularity violations in cyber‐physical systems. Systems Engineering, 23(3), 338-349.
3. Mo, R., Cai, Y., Kazman, R., Xiao, L., & Feng, Q. (2016, May). Decoupling level: a new metric for architectural maintenance complexity. In 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE) (pp. 499-510). IEEE.
4. Baldwin, C. Y., & Clark, K. B. (2000). Design rules: The power of modularity (Vol. 1). MIT press.
5. Bijlsma, D., Ferreira, M. A., Luijten, B., & Visser, J. (2012). Faster issue resolution with higher technical quality of software. Software quality journal, 20(2), 265-285.
6. Browning, T. R. (2001). Applying the design structure matrix to system decomposition and integration problems: a review and new directions. IEEE Transactions on Engineering management, 48(3), 292-306.
7. McCabe, T. J. (1976). A complexity measure. IEEE Transactions on software Engineering, (4), 308-320.
8. Bouwers, E., Van Deursen, A., & Visser, J. (2014, September). Quantifying the encapsulation of implemented software architectures. In 2014 IEEE International Conference on Software Maintenance and Evolution (pp. 211-220). IEEE.
9. Vierlboeck, M., Dunbar, D., & Nilchiani, R. (2022, April). Natural Language Processing to Extract Contextual Structure from Requirements. In 2022 IEEE International Systems Conference (SysCon) (pp. 1-8). IEEE.
10. Vierlboeck, M., Lipizzi, C., & Nilchiani, R. (2022). Natural Language in Requirements Engineering for Structure Inference--An Integrative Review. arXiv preprint arXiv:2202.05065.
11. Zhao, L., Alhoshan, W., Ferrari, A., Letsholo, K. J., Ajagbe, M. A., Chioasca, E. V., & Batista-Navarro, R. T. (2021). Natural language processing for requirements engineering: a systematic mapping study. ACM Computing Surveys (CSUR), 54(3), 1-41.
12. Liddy, E.D. 2001. Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc.
13. Rajkumar, R., Lee, I., Sha, L., & Stankovic, J. (2010, June). Cyber-physical systems: the next computing revolution. In Design automation conference (pp. 731-736). IEEE.
14. Bouwers, E. M. (2013). Metric-based Evaluation of Implemented Software Architectures.
15. Bouwers, E., Correia, J. P., van Deursen, A., & Visser, J. (2011, June). Quantifying the analyzability of software architectures. In 2011 Ninth Working IEEE/IFIP Conference on Software Architecture (pp. 83-92). IEEE.