Individual Research Paper

ENGN2225 Systems Engineering Design

Subsystem Integration for a Degree Planning System

Nilupuli Senadhira
u5194333@anu.edu.au
Tutorial Group- Wednesday 3-5pm W15R

Abstract

Subsystem integration is a concept in Systems Engineering which assists in determining the systems architecture and in defining the boundaries and limitations of a system design. In this paper, design problem of a Degree Planning aid for ANU is used to demonstrate this concept.

The aim of this paper is to breakdown the degree planning system into components in order to establish the systems architecture. A top down allocation approach is used to categorize each element into subsystems, establishing interactions between subsystems (endogenous variables) and external variables of the overall system within a system boundary. This is achieved by constructing a System Boundary Chart and a System Interface to determine the design architecture.

Background

Subsystem Integration concept background

The Subsystem Integration approach consists of several steps, starting with System boundary chart, Functional Block Diagram and System and subsystem diagram. The overall process defines the scope and the systems architecture of the solution to the systems engineering problem Edwards in his paper defines a system, to be consisting of a large number of subsystems each of which performs a different function, interconnected to perform a common purpose. In his paper , he emphasizes the importance of defining the system boundary for an effective design and implementation of the system(Edwards, 1990). One of the most efficient methods in systems engineering to summarize the scope is to create boundary charts which categorizes the system components into Endogenous, Exogenous and excluded variables (Sterman and J.D., 2000).

Blanchard claims that a system should be disintegrated into packages grouping the similar elements based on aspects such as functionality etc. through partitioning. The author suggests several objectives to be considered in constructing a packaging scheme. He emphasizes that the interaction of a subsystem with other subsystems should be kept minimum while maintaining maximum internal complexity, as this allows the maintenance of the system without massive changes in functionality(Blanchard, 2011). The system partitioning then make up the overall systems architecture. Proper systems architecture guides to a solution which satisfies the maximum customer requirements evaluated using the House of Quality and helps to incorporate them into functional analysis defining the subsystems, resource requirements and variables in the system(Blanchard, 2011, 2014) .

Leonard defines the 'Design synthesis' as the design product based on the functional analysis(Leonard, 1999). This develops a physical architecture that satisfies the functional and performance requirements within the system limitations. The authors states that, 'The objective of design synthesis is to combine and restructure hardware and software components in such a way to achieve a design solution capable of satisfying the stated requirements'(Leonard, 1999). The author also emphasizes that it is required that each physical or software component should meet at least one functional requirement defined in the system, and also the performance among parameters are tracked using metrics (Leonard, 1999).

According to Leonard, the desirable characteristics of a system interface includes, low coupling in order to favour easy decoupling in case of later modifications, high cohesion to use similar components in order to perform multiple functions and low connectivity among subsystems to reduce the system complexity(Leonard, 1999). This stage of the system engineering approach is beneficial because of the scalability it provides by enabling to adapt the depth of the details of the model to the design problem(Weilkiens, 2011).

Case study

An example of an FBD of a solid-state Radiant Energy Management System is provided in the Appendix. According the FBD, the dotted line defines the system boundary. The internal variables of the design architecture includes shock charging power circuitry, microcontroller interface circuitry, xPOD head assembly, microcontroller and LCD display, whereas the external variables comprise of external supply, output load, internal supply and the two battery banks. However it's not possible to determine the scope and limitations of the system, since excluded variables are not included in the standard FBD.

The first battery bank powers the controller and the load whereas the shock charging power circuitry is being powered by an external supply. In addition, magnetization/demagnetization data will be sent to the shock charging power circuitry. This excess electrical energy will be stored in the second battery. This exemplifies the application of the concept of subsystem integration in a real life design problem

Project background

This paper aims to perform the application of the concept of the 'system integration' for a degree planning aid for ANU. The project is mainly focussed towards the students enrolled in undergraduate engineering degrees. Due to the modification of the rules based upon the year of enrolment, students find it difficult to satisfy the degree requirements in order to graduate on time. In this research paper, we are using a systems engineering approach to solve the above problem. Before the application of systems integration, the design requirements of the problem were determined using HoQ method and then systems analysis was performed using FFBD. During the concept generation, the group decided that the best solution for this engineering problem is a feedback rich software system which generates a degree plan as the final output. The purpose of this paper is to partition the proposed system into components, defining the individual subsystems with high internal complexity, minimizing the external interactions with other packages (Blanchard, 2011). This process will be accomplished using a System Boundary Chart which will be translated into a System-Subsystem Diagram via a Functional Block diagram.

Systems boundary chart for the degree planning system

Table 1 below illustrates the System Boundary Chart for the proposed degree planning aid. The Endogenous variables include the main subsystems of the overall system while exogenous variables define the inputs and outputs, and the aspects that are extended beyond the scope of the design are categorized under excluded variables.

System boundary Chart		
Endogenous	**Exogenous**	**Excluded**
interface	user	IT cost
database	feedback	changes in degree requirements
sever module	real-time input	other colleges
memory	real-time output	course changes(majors, minor, electives)
	type of the degree	availability of academics
	degree preference	timetabling issues
	commencement year	career types other than undergraduates
	degree plan	

Table 1-boundary chart for the degree planning system

The internal variables of the design architecture consist of the subsystems of a general software system. These subsystems will be divided further into components in producing the FBD. Kapurch claims that, when the element becomes more delicate, the proper sequence becomes more critical. As a result, a small change can cause a considerable impact on the functionality and the performance of the design (Kapurch, 2010). Since the proposed degree planning aid, is supposed to be a dynamic and a feedback rich design, fine partitioning of the endogenous variables assist in defining the optimal resource requirements and design architecture.

The exogenous variable of the software system includes the input and outputs to the system. The main inputs to the system are user and the feedback. Feedback includes the real-time inputs and real-time outputs of the system. The final user output will be a complete degree plan which satisfies all the degree requirements.

The external variables are listed in table 1 determine the boundaries and the scope of the system. In defining the proper design architecture, it is important to be aware of the existence of the system boundaries and recognize the limitations of the system. For an instance, the proposed degree planning aid is limited to the undergraduate Engineering degree planning. Furthermore, the issues like course changes and changes in the degree requirements are also excluded from the system, since such issues are handled by the ANU Curriculum Development Committee. Most importantly, the IT cost is also excluded, because in the process of requirements engineering, ‘low development cost’ was ranked as one of the lowest design requirements and also it was decided, that the low maintenance cost would help to redeem the initial investment. The excluded variables provide a foundation in improving and extending the scope of the design. For the degree planning software system a future improvement might be, to create a program which is capable of generating degree plans for all the career types and degrees in all the ANU colleges.

Functional Block Diagram for the Degree Planning System

According to Weilkiens, it’s important to analyse the type of embedding for a proper integration system and surrounding to prevent negative consequences in the performance(Weilkiens, 2011). An elaborate system interface for the proposed ANU degree planner system is constructed by generating an FBD as shown in figure 1. The dotted line is the system boundary and surrounds the system blocks. In addition, all the currently known external variables are denoted all around the system and associations are used to link them(Weilkiens, 2011).

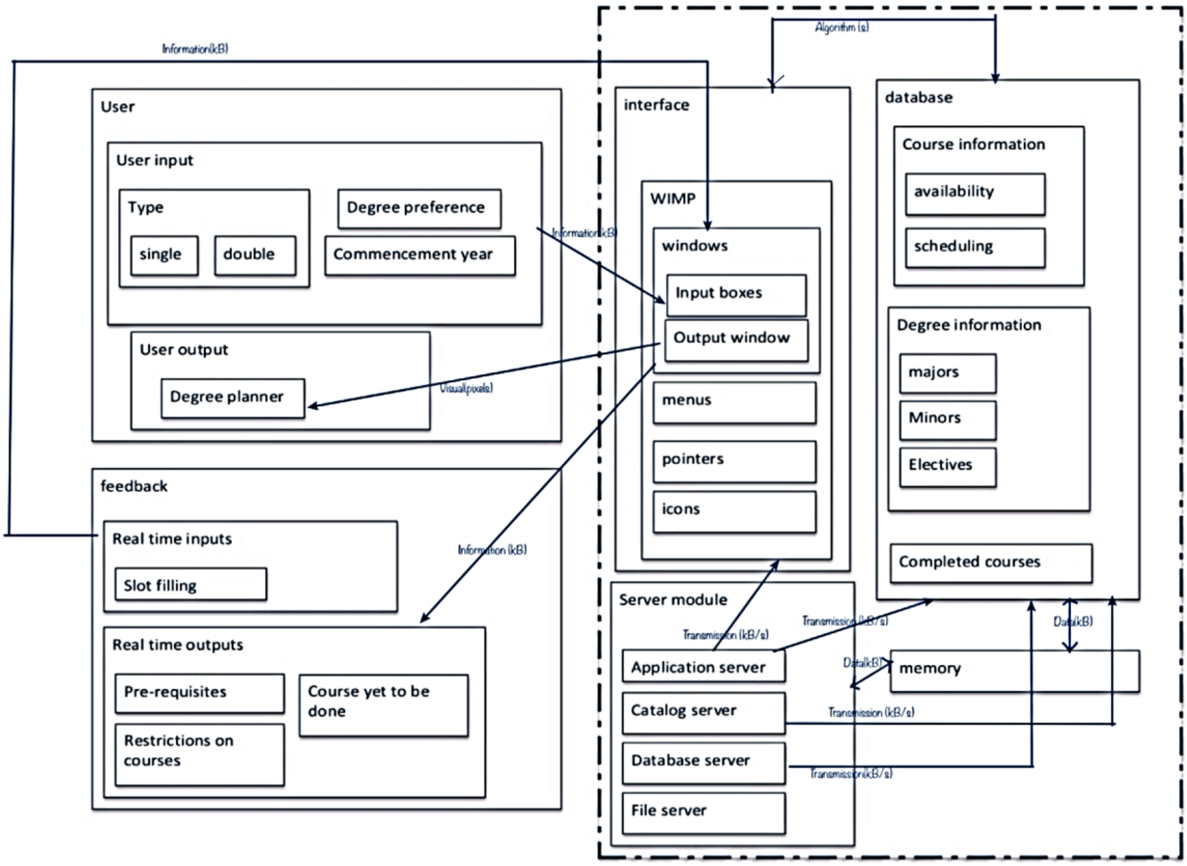

Figure 1-FBD for the degree planning system

The design architecture consists of 4 subsystems which are partitioned further. User and feedback compose the exogenous variables. As shown in figure1, user is further categorized into user input and user output whereas real-time inputs and outputs are included in the feedback.

As shown in Figure 1, user and feedback are considered as two different external variables. This is based on the fact that 'user input' inside the user block refers to the initial fixed inputs and the 'read-time inputs' in the feedback refers to the dynamic input entered by the user as he/she plans the degree using the program. Similarly, the 'user output' in the user block refers to the ultimate degree plan delivered to the user while the 'real-time outputs', generates error message regarding uncompleted courses and pre-requisites.

The user inputs such as degree preference, commencement year and the type of the degree will be provided to the system via the input boxes in the interface. The interface communicates with the database via algorithms. The database of the software holds information about completed courses and other degree information.

Since the system is a feedback rich design, information is communicated simultaneously with the windows of the interface and the real-time inputs and outputs. The proposed system will be designed such that, the slots of the degree plan is able to fill in real-time checking the course availability, relevant pre-requisite and course restrictions. If the selected course doesn't satisfy at least one of the above criteria, error messages will be generated to inform user about the issue. This option will be useful in choosing elective courses.

The memory plays an important role in the systems architecture, since it stores both temporary and permanent variables/parameters created during the process. This data is communicated to both database and the server module on demand. The server module is basically the heart of the design as it serves the requests of the other subsystems. Application server is a part of the server module which executes procedures hence connected to the interface and database. Catalog server accesses database while the database server provides database services to the rest database subsystem. The file server performs the storage and backup functions in degree planning system. Once server module performs all the database and application procedures, signals will be transmitted to the interface generating the final degree plan.

Among the subsystems indicated in the system interface, database and server module can be regarded as predominant subsystems. The database contains information about the available courses and degrees. When designing the degree planner, it is important to focus more on this subsystem as the pairwise ranking of design requirements carried out at an earlier stage indicates that providing students with accurate information should be the foremost design requirement addressed in this design solution. It is also important to ensure the reliability and updatability of the database information and this can be achieved by associating ISIS, wattle, Study at and Programs and Courses with the proposed degree planner software.

Conclusion

A systems boundary chart was constructed in order to identify the internal and external variables and also to determine the limitations and scope of the design. These identified variables were used to generate a FBD for the proposed ANU degree planning system and the application of this concept will help the group to produce an appropriate design architecture which satisfies the needs of the stakeholders while defining the resource requirements of the design.

Bibliography

2014. *Sysarch.pbworks.com* [Online]. Available: http://sysarch.pbworks.com/w/page/7241231/FrontPage [Accessed 31 March 2014].

BLANCHARD, B. S., W.J. FABRYCKY 2011. Systems Engineering and Analysis. *Functional Allocation.* Pearson, New Jersey.

EDWARDS, L. Beating the bounds [defining systems engineering boundaries]. Systems Integration: Principles and Practice, IEE Colloquium on, 1990. IET, 2/1-2/3.

KAPURCH, S. J. 2010. *NASA Systems Engineering Handbook,* DIANE Publishing.

LEONARD, J. 1999. Systems engineering fundamentals. DTIC Document.

STERMAN & J.D. 2000. The subsystem diagram-application to a semiconductor firm. *Systems Thinking and Modelling for a complex world.* McGraw Hill.

WEILKIENS, T. 2011. *Systems engineering with SysML/UML: modeling, analysis, design*, Morgan Kaufmann.

Peer review Critique

I received only one peer review for my draft.
Peer review 1:
Aspect1: The reviewer has commented that I haven't included the system-subsystems diagram, discussion and conclusion. The author had only stated about the things that I have done and need to be done in the review. So I didn't find this comment helpful at all. However I have included the missing parts in the final research paper.
Aspect2: The author states that, I have to go deep into the creation of the system-subsystem diagram and how the internal and external variables are related to the system and to my project. Since I didn't include a FBD of the degree planning system, I was unable to demonstrate the theory related to system architecture properly in the draft. But I have included a FBD and a proper discussion in the paper.
Aspect3: I received an outstanding grade for this aspect, so I didn't make any major changes on the theory.
Aspect4: The author states that I need to extend the project to various other subjects. He also states that I have only focused on CECS. However, I have mentioned in the System boundary chart that other colleges will not be considered within the scope of the project and have categorized it under excluded variables. The reviewer also states that I could include requirements from different colleges and observe the changes in the subsystems. Although, I think this is not necessary because I have clear stated the limitations of the design in the research paper and variation in subsystems and (design) requirements could be analyzed in the attributes cascade.
Aspect5:The reviewer mentions that the bibliography was not done in Harvard formatting and 'citations seem a little sloppy'. I accept the fact that the citations were bit unsystematic, but referencing was done in Harvard format. I have organized the citations appropriately in the final research paper and I found this comment was helpful in formatting the document.

Tutor feedback

My tutor Yimeng Jiang, provided me with some helpful comments regarding the FBD. She told me to change the association between database and interface, so that algorithms will be communicated both ways, and I have changed the diagram accordingly in my paper. She also mentioned that, the feedback block should be inside the user block, and having them as two main external variables make the system redundant. Since I received the comment a day before the due date, I couldn't merge those two variables into a single block. However I have commented on this in the FBD analysis/discussion (the reason for having user and feedback as two different variables) and the correct version of the FBD will be used in the attributes cascade and in the final report.

Appendix

Case study