# Question framework enhancements

## Question framework

Problem framing is a problem-oriented modeling approach, which is primarily used to analyze and solve software problems. In problem framing development, problem analysis is often combined with the description of the problem context – the realm in the real world that forms the context of the solution sought – and the description of the requirements – the change in the context of the problem that the solution should bring.

Jackson introduced the Problem Framework [2] as a framework for analyzing and cataloguing software problems. In their initial formulation, they focused on the specification of the software solution. The problem framework is the concretization of the ideas of Michael · Jackson et al. in separating machines from descriptions of their environments. This separation is often considered a useful principle for needs analysis.

As shown in Figure 1, a simple example of the problem diagram is illustrated. In the problem diagram, the dotted ellipse represents the demand, and here the Stop light and Go light should be controlled according to the determined law. Attach the ellipse to the dotted arrow on the light group

A requirement reference is not an interface to a shared phenomenon, and the requirement itself is just a description that is generated during the development process, not a domain. In the context of the problem, it has no physical implementation, so it cannot share the phenomenon with machines or any other domain. But we still represent and label it in the problem diagram in much the same way as we represent interfaces.

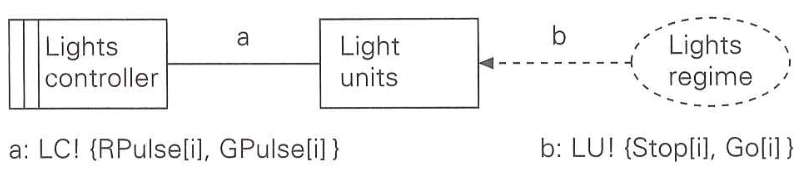


Figure 1: An example of a problem diagram

Modern organizations face the challenges of the times of volatility, uncertainty, complexity, and ambiguity[1] has led to the need for many organizations to undergo significant organizational changes to survive. Therefore, in organizational structure management, there is a widely accepted view that organizational structures need to be built that are flexible enough to adapt and adapt to changing external environments and internal needs. However, in the framework of the problem, there is a lack of clear treatment of organizational change and evolution. This study considers how to apply problem framing in the context of complex socio-technical system design, and discusses the implications of the problem framing basis and symbolic aspects.

In this section, we briefly review the basics of problem framing and summarize the elements of our approach, focusing on the structure of the organization, and proposing new ways to characterize the structure of the organization. Our approach is based on the Jackson Problem Framework, a well-known representation and analysis framework for software problems. We've expanded the framework to provide a symbolic representation of organizational structure representation and analysis. We look at this approach with a case study from a real-world perspective.

## Question framework enhancements

A problem framework is a general framework for representing and analyzing problems, using elements such as domains, activities, and problems to define and classify problems. The Jackson Problem Framework is commonly used in software engineering to help understand and manage complexity by representing system requirements and design constraints, as well as by analyzing interactions between domains.

The advantage of a problem framework is that it clearly defines the problem and can identify the root cause of the problem, thus helping managers and teams to better respond to the problem. However, there are some limitations to the problem framework when applied to organizational structure management. While it can be used to describe and analyze problems, it does not provide sufficient tools to understand and deal with the complexity and dynamics of organizational architecture.

Therefore, our team has extended the problem framework to accommodate the analysis of issues related to the organizational structure.

Since there is a hierarchical structure in the organizational structure, in order to facilitate the distinction between different levels, and to improve the ease of use of the model, we use thicker borders and fonts for the higher-level organizational domains, and make the borders and fonts of the same level of organizational domains consistent with the thickness of the font, and the included levels will be represented by thinner borders and fonts. As shown in Figure 2, Company C is the highest level that includes the algorithm team, Company C uses the thickest border and font, the algorithm team uses the thinner border and font as the second level, and the algorithm engineer and AI architect are both at the lowest level and use the thinnest border and font, and they are connected by black dotted lines.



Figure 2: An example of a hierarchical inclusion relationship

In addition, it is difficult to represent the communication between organizational domains in the traditional problem framework, so we have added three different lines to represent different communication situations: a solid blue line for correct communication, a red dotted line for lack of communication, and a solid red line for abnormal communication.

Organizational Domain. Organizational domains are used to represent the hierarchy of abstractions, where there is only an inclusion relationship between the organizational domain and other domains, and the organizational domain is not associated with the phenomenon of sharing. Figure 3 illustrates the legend of the above extension.

Figure 3: Extended legend

## Case Study:

In the case studied in this article, a major global software development company is working with another company responsible for requirements analysis to develop a piece of software. In turn, the company responsible for requirements analysis needs to complete the collection, analysis, cleaning and labeling of requirements. Software development companies and requirements analysis companies need to work together to complete their respective tasks and ultimately deliver the software to investors. As the project progressed, there was a discrepancy in the communication between the two companies. Through this failed cooperation, the two companies hope to make the flaws in the organizational structure more intuitive, so that the project can be avoided by analyzing the organizational structure at the beginning of the project.

Based on the above examples, we provide a modeling approach for the organizational structure. Figure 4 illustrates this approach, where a solid red line indicates abnormal communication, a red dotted line indicates a lack of communication, a solid blue line indicates correct communication, and a dotted line connecting domains indicates an inclusion relationship.

The field of demand analysis represents demand analysis companies;

Representing software development companies in the field of software development;

Clients represent investors;

Figure 4: Problem Framework Extension: Organizational Structure

In this case, the IT department of Company A was supposed to communicate with the CTO of Party B, but because the IT department did not summarize its own opinions and needs during the communication, it was unable to effectively express its views and needs in the communication with the CTO, resulting in communication failure. To avoid this happening again, IT departments should organize their opinions and needs in advance and express them clearly and clearly in communication.

By marking the lack of communication, it is more intuitive to derive from the model which part of the communication is necessary but missing, and the case of abnormal communication needs to be understood in more detail, and different improvement strategies can be adopted according to different abnormal situations. Proper communication can be used to see what departments are connected to each other.

## Zoom-in

In the task-driven zoom-in diagram, you can start from the task to analyze the problems existing in the actual task execution process of each department, so as to view the key people and the scale of the task. After zoom-in, the high-level structure will be weakened, the lowest level of individuals will be magnified, more focused on communication between people, in the correct communication channel all communication channels should be blue solid line, and when the communication is found to be missing (red dotted line), it is necessary to correct and adjust the communication process between organizations in time. The picture after Zoom-in is more convenient for people in various positions to clarify the specific individual who is connected with them when performing the task, as well as the specific channel of the whole process, and have a certain grasp of the scale of task resources and task-related departments, and the prediction of the task duration can also be more accurate.

 Figure 5: Zoom-in diagram

After zoom-in, a path will be formed from the task as the starting point and the completion of the task as the end point, and the nodes on this path are the people involved in the task, and the personnel are stored in the department. As shown in Figure 5, this is a data layer development task and the various personnel involved in completing this task will become nodes in the task path, and the task is generated by Company A's needs, so the starting point is Company A, initiated by Company A's IT department, data scientist and innovation department, the IT manager in the IT department also needs to report the work progress of the IT department to the project manager of the innovation department, and the data scientist also needs to provide the obtained data to the project manager, and the operation A, B collects the real data in the community and summarizes it to the project manager, and the channel of the task in Company A is completed. It is not difficult to see that the key person in Company A is the project manager, and it can be said that the project manager of Company A plays a decisive role in the success or failure of the task. For Company B, the data field expert needs to collect and analyze the data and summarize the information to the CTO of Company B, and the CTO communicates with Company A's project manager to complete the task of communication on Company B. It can be seen that the person in charge of Company B is the CTO, and their responsibility is to process the data and feedback the results of the processing to Company A. For Company C, the data layer development requires the combination of front-end and back-end development, and they summarize the information to the pre-sales, and the product manager also needs to communicate the information of the product with the pre-sales, and then summarize the information to the project manager of the B-1 side before the sales, and B-The project manager of Party 1 will collect information from the algorithm team and the engineering team, summarize it to the project manager of Company A, and the task will be completed in the channel of Company C.

To sum up, after zoom-in, the overall process of the task can be analyzed more clearly, the details of communication can be retained, the problem can be quickly analyzed when there is abnormal communication, and the communication channel of the organizational structure can be improved in time by accurately locating the problem and improving the success rate of the project.

## References、

[1]Huang Shan,Wang Tao,Zhang Zhongshan,els. Analyzing Hierarchical Organizational Structure Change: Strategic Planning Model from Personnel Perspective[C]//2023 IEEE International Conference on Data Mining Workshops (ICDMW): IEEE, 2023.

[2] Jackson, M. (1997). The Meaning of Requirements. Annals of Software Engineering, Vol 3, Pp5-21, Baltzer Science Publishers.