

Lecture No. 7

3.8 Source and sink analysis

Once requirements are documented using any of these analysis models, an independent verification is needed to verify completeness and consistency of requirements captured through these models. The process of verifying requirements involves careful analysis of sources as well as the sinks of information.

Source

A stakeholder describes requirements (needs, constraints) to be included as system functionality. These can be processes that generate certain information that the system may have to process or maintain. Sources of requirements are the origins from where the corresponding business process is initiated. By this concept, one has to trace from a requirement back to its origins to see who is involved in its initiation. Be it a person, an organization or an external entity that initiate some action and system responds back by completing that action.

Sink

Sink is the consumer of certain information. It is that entity which provides a logical end to a business process. Thus, 'sinks of requirements' is a concept that helps in identifying persons, organizations or external systems that gets certain functionality from the system. These are logical ends of requirements, or where all the requirements are consumed. For example, we may consider a user of a software application that retrieves a report from the system. In this case, user when reviews the report, becomes the sink of that report. Thus when analyzing the sink of the requirement of implementing a report, the analyst would naturally point towards the user who would get that report.

In source and sink analysis the analyst determines all the sources of requirements and where do these requirements consume (sinks). Now evaluate a report which displays certain information, the source of this report is the data (and who enters it) that is input to be retrieved later in the form of the report. Similarly, whoever needs this report become the sink of the report.

In a similar manner, at times we gather data in our application that is not used anywhere. So the question really is what to do with that kind of unused data or the missing requirement. Is it really redundant or is something really missing from these requirements? How to figure it out?

For example, we are having certain inputs (sources) to a process against which we do not know about the corresponding outputs (sinks). Such inputs are redundant if there is found no corresponding outputs. Thus these inputs can be removed as redundant. If we probe out corresponding outputs, which could not be recorded initially, that mean these inputs were not redundant rather a few (output related) requirements were missing that we discovered during the sink analysis.

A stakeholder may have required the development team to develop certain report for his use. It means we are sure of its use (sink) but not about its sources, from where the required information will be provided? Who will input that information and using what mechanism?

A requirement statement that describe the report but do not list down its sources, will be an incomplete statement and the software engineer who is involved in validating such requirements, should identify all the sources against sinks or vice versa to determine complete end-to-end requirements.

Process Models

Domain Models

During requirements analysis phase, different models are developed to express requirements of the system. Though it is difficult to draw a line between these models as they complement each other, they differ in the manner information is expressed in these models. Most of these models are pictorial and contain explanation to the diagrams. Some of these models are discussed in the following subsections.

Understanding the business domain

It must always be kept in mind that the first step in delivering a system is establishing what needs to be driven. That is, clear understanding of the problem domain is imperative in successful delivery of a software solution. A software developer has to develop an understanding of the business problem he is trying to solve. Unless he develops this understanding, it is really difficult, if not impossible, to develop the right solution. But at least if he collects both ends (sources, sinks) involved in different processes of the business system, the corresponding requirements will be complete and yield a better understanding of the problem domain. A software engineer works on domains that may not correspond to his field of specialization (computer science, software engineering). He may be involved in the development of an embedded application that automates the control pad of a microwave machine or a decision support application for a stock exchange broker. As the underlying systems for which these software applications are being developed are not software systems, the software engineer cannot be expected to know about these domains. So, how should he get all the required knowledge about these systems? As without acquiring this knowledge, he may not be able to write down complete and unambiguous requirements which are acceptable to users as well.

An important difference between software and another engineering discipline is that the software engineer has to work on problems that do not directly relate to software engineering. Whereas, an electrical engineer will work on electrical domain problems, a civil engineer will work on civil engineering problems and so on. So, software engineer has to learn user vocabulary and terms which they use in their routine operations. To overcome this problem, a number of domain gathering techniques are used. These techniques help in extracting requirements from systems which are not known to a software engineer. Using these techniques the requirements gathering and validation process becomes convenient and manageable for a software engineer.

The following subsections discuss some of these techniques.

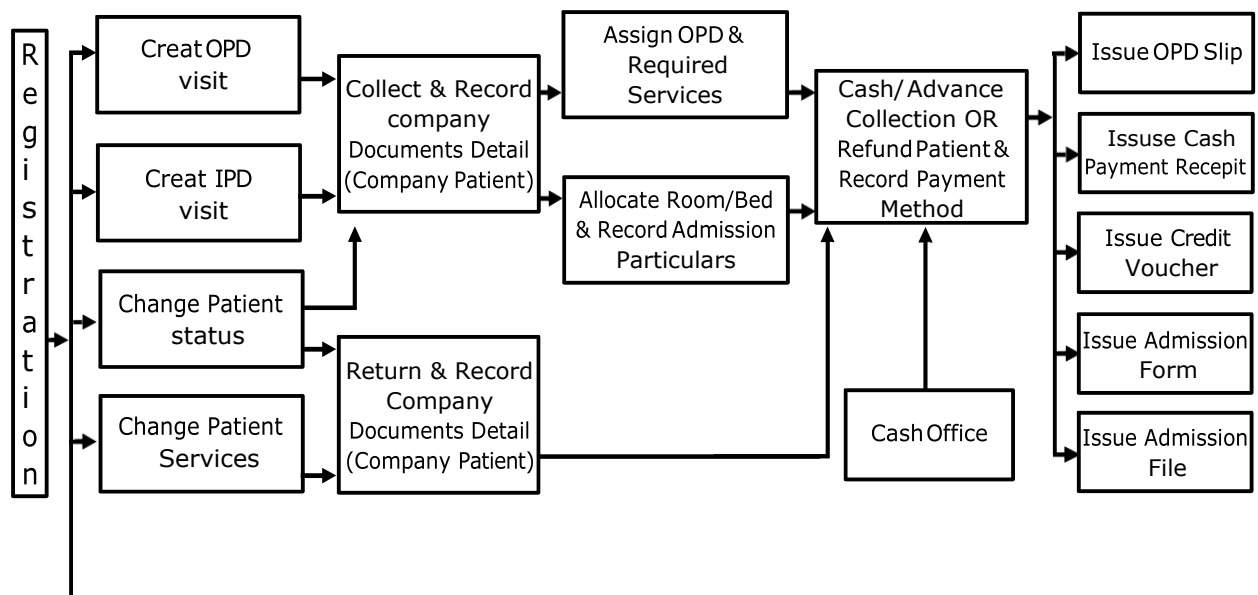
4.2 Logical System Models

System models are techniques used to understand user needs and software engineer use these techniques in order to understand business domain. Software engineers develop diagrams to model different business processes. System models include the following

- User business processes
- User activities for conducting the business processes
- Processes that need to be automated
- Processes which are not to be automated

Business process model

The first model that we will look at is called the process model. This model provides a high-level pictorial view of the business process. This model can be used as a starting point in giving the basic orientation to the reader of the document. Following is an example of a hospital registration system which deals with two types of patients.



As opposed to flow charts, there are parallel activities in this diagram which are further elaborated by specifying their major activities. The process described in this diagram is as follows

- A patient may come to visit In Patient Department (IPD) or output patient department (OPD)
- System determines if he is a company patient or a private patient.
- For a company patient, system verifies him.
- For an OPD patient, system will issue a chit to the patient and inform him about his number and the consultant to whom he has to consult and he will have to wait for his turn.
- After verifying an IPD patient, system will create a visit and allocate him a room or a bed etc. If system cannot allocate this, then it will inform the patient. Otherwise the patient is checked in and his information is maintained in the system.

- System displays information about the expenses of the required service to the patient so that he is informed of his expected expenditure.
- Some advance payment is also received against the required service and this amount is adjusted in the final settlement.
- All this information is supplied to cash office that eventually deals with payments, etc.
- Upon receiving the cash, for OPD patient, a chit will be issued. For IPD patient, an admission form will be filled and this information will be maintained in the system. A receipt will be issued to the patient.
- For credit transaction, corresponding voucher will be prepared.
- So the model depicts process before the start of the treatment.
- A patient may ask to change his service on event of an unsatisfied response from the hospital staff or any other reason. System may cancel his record and pay his amount back.
- Similarly, a doctor may ask a patient to change his status from OPD to IPD.

In a business process diagram, following points are important and should be noted

- It does not describe the automated system
- It only reflects the existing process of the user to help software engineer/analyst in understanding business domain.
- It may contain information on processes that need not be automated.