

3.2 Context diagrams

What is this?

A diagram of the product and its surroundings.
It shows the scope of the product.
It is important but often overlooked.

The context diagram (Figure 3.2) shows the product as a black box surrounded by user groups and external systems with which it communicates. Arrows show how data flow between the product and its surroundings. In the figure, you see that the hotel system communicates with receptionists about booking, checking out, recording service charges, etc. The hotel guest receives a booking confirmation and invoices printed by the system. There is an external accounting system that receives account records corresponding to the printed invoices. There is also a telephone system (a local exchange) that controls the hotel room phones and sends billing information to the hotel system.

The notation is a kind of dataflow diagram. It should be supplemented with a plain text explanation of each interface.

The arrows show transfer of data. For instance, the diagram on the figure shows that the receptionist exchanges booking data with the product. However, sometimes analysts let the arrows show events rather than data. The arrow then shows who takes the initiative. For instance, there is an arrow from the hotel system to the accounting system, indicating that the hotel system takes the initiative to transfer data to the accounting system. Data may actually flow both ways during the transfer, because the accounting system sends various confirmation data back to the hotel system. A plain text explanation could help to clarify these matters.

Requirements. The context diagram gives an excellent overview of the required product interfaces. It reflects an early decision about the scope of the product: what is included in the new product and what is outside (domain). The example shows clearly that the accounting system and the telephone system are not included in the hotel system in this project.

You can use the context diagram as a requirement, for instance in this way:

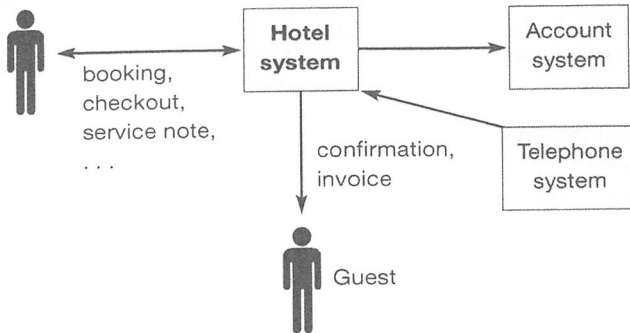
R1 The product shall have the interfaces shown on the context diagram.

You might help the reader by outlining also the domain, as shown at the bottom of Figure 3.2. This is not a requirement specification, however.

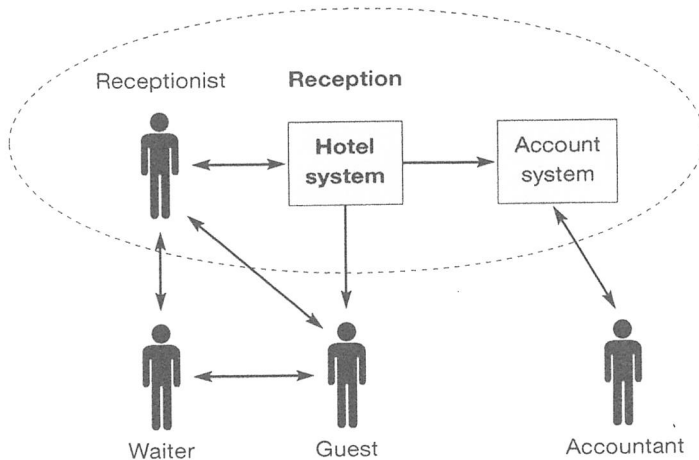
Fig 3.2 A context diagram

R1: The product shall have the following interfaces:

Receptionist



R2 ??: The reception domain communicates with the surroundings in this way:



Sometimes analysts show data flowing between the surrounding users and products. As an example, we have shown that the guest communicates with the waiter, and the receptionist. This has nothing to do with the product requirements, but may help the reader understand the context.

Practice. It is extremely useful to outline a context diagram early in the project and keep it updated during analysis. Making the diagram can reveal amazing differences in the understanding of what the product really comprises. For instance, what do we mean by a reception system? Does it include the telephone system?

Should we require both systems as one package? Does it communicate directly with the waiter or only with the receptionist?

Surprisingly, we rarely see context diagrams in larger requirement specifications. None of the specifications in Chapters 11 to 14 have any. Many external systems were mentioned in these requirements, but it was difficult to see to what extent the product communicated with them – or whether these systems were part of the required package. A context diagram would have been very useful.

The membership system in Chapter 15 has a verbal version of a context diagram and requirement R1 says that the system shall support these interfaces. (Actually, the analysts had sketched a diagram but included only the text version in order to save time.)

Advantages of context diagrams

Verification. The context diagram gives developers an over-view of the interfaces and serves as a high-level checklist over what to develop. It is easy to verify that all interfaces are dealt with during and after development.

Validation. Most customers can readily understand the context diagram, spot missing interfaces and discuss what is to be delivered and what is outside the product.