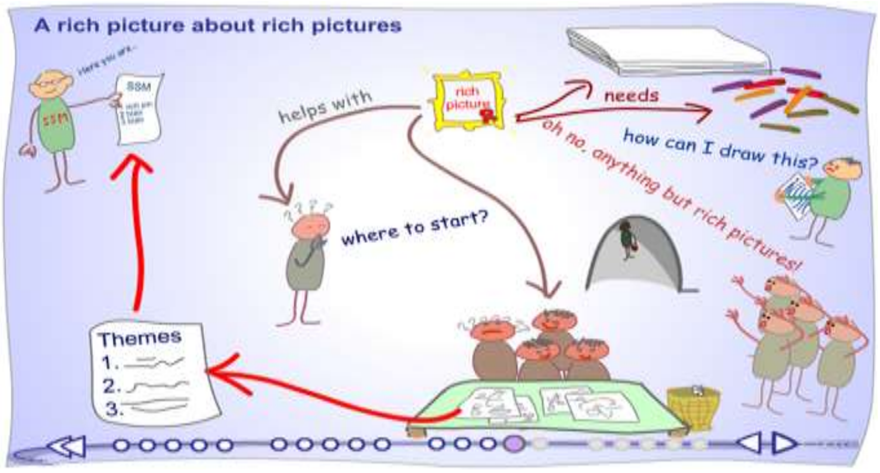
Introducing Rich Pictures

A rich picture is an effective tool for analysing problems and expressing ideas. For example, given below is a rich picture about how to draw rich pictures.



When developing a solution to a business problem, it is essential to understand the vital components of that problem. Rich pictures can help you to identify:

- Business processes and their data requirements
- The actors involved in the processes and their responsibilities
- The relationships between processes and actors
- Potential problems and conflicts

For example, given below is a rich picture of a web-design consultancy.

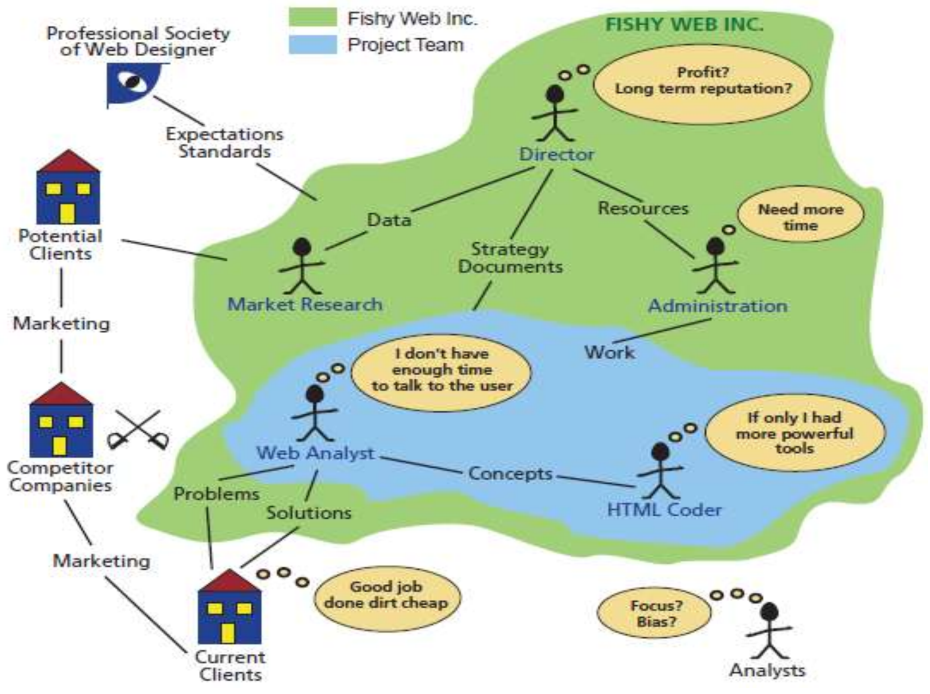


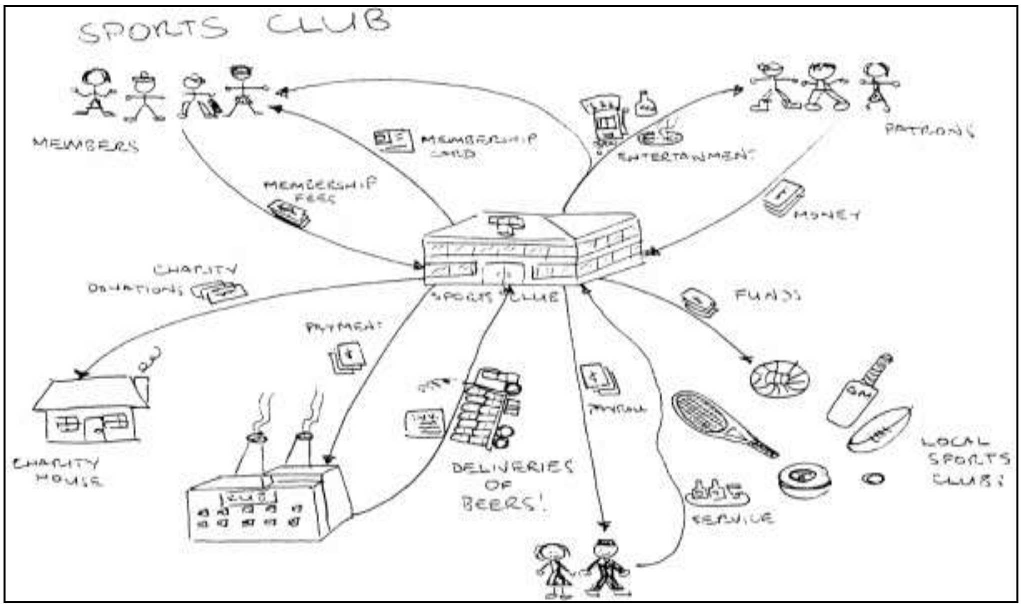
Figure 2 Rich Picture of Web Design Consultancy

How to Draw a Rich Picture?

One approach is to start with a short problem statement, in the centre of the page; and then, place all the relevant keywords around it.

For example, take the running of a sports club. What keywords are likely to be relevant? The keywords that you come up with will prompt you to consider those issues. To start with, write down everything you can think of. Later on, you can remove those keywords that you decide are irrelevant.

Consider the sports club rich picture given below.

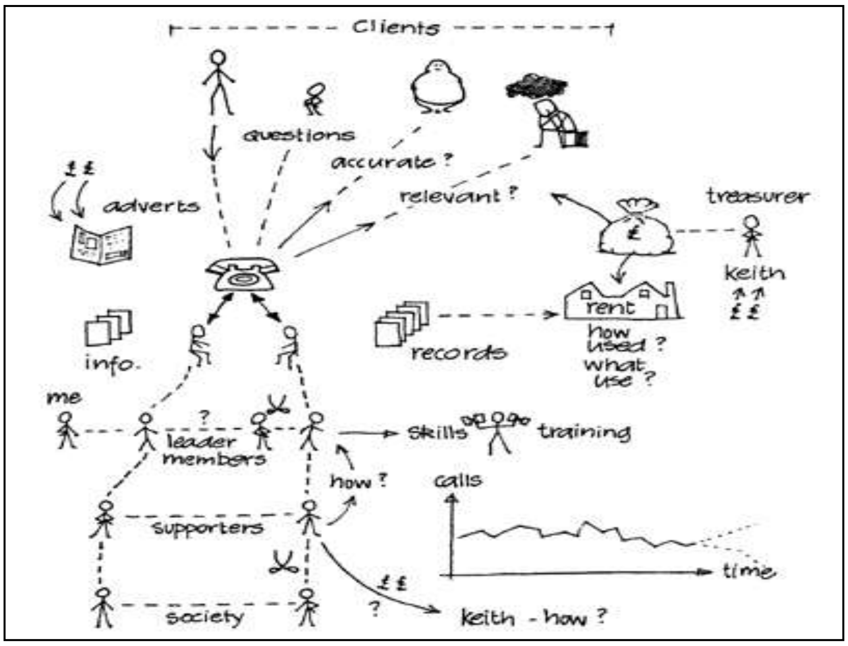


As another example, consider the computerisation of a “tool hire” company. Your keywords might be tools, borrowers, and staff. The keyword ‘tools’ should prompt you to consider different categories of tools, for which the keywords could be things like drills, lathes, and ladders. Each such keyword should, in turn, prompt you to consider different types of tools in each category. And, so on.

Your rich picture should contain all the relevant keywords. However, it is the use of pictures and diagrams to represent concepts and relationships that makes rich pictures so popular. This is because a good, clear picture will communicate ideas more readily than words. So, d*o not make your rich picture too wordy.*

Also, do not expect to complete your rich picture straightaway. Start with a rough version of the problem domain and develop it over time. As you develop your rich picture, you can identify further issues to consider, as well as what keywords to include and what to reject.

Shown below is an incomplete rich picture.

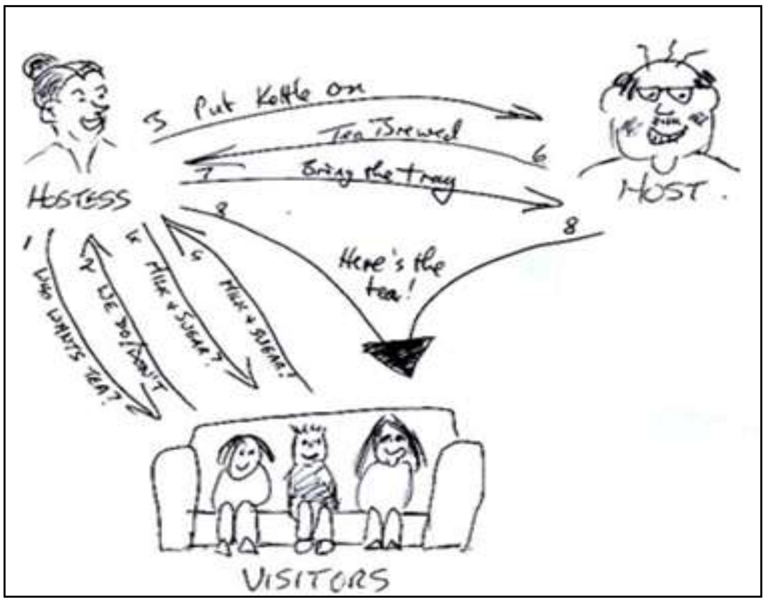


Not every rich picture can convey its message effectively.

If you cannot understand what a rich picture is saying, then it is either incomplete or it has been drawn badly.

Discuss with your team mate how effective you think the rich pictures in this document are. Be critical. What improvements would you make?

Finally, for a little light relief, here is a rich picture of how to entertain visitors:



You will have realised by now that there is no standard notation for drawing rich pictures.

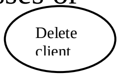
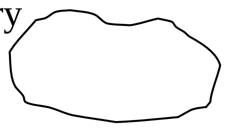
You could make up your own notation, if you wanted to. However, it is always good to have some kind of standard.

Given below are the guidelines that this module expects you to follow.

Your rich picture must tell a story. This means using images, pictures, keywords and descriptive labels, to give the reader a very good idea of what is going on. In terms a business problem, your rich picture must say who is processing what data for what purpose, what data is coming into the system, what information is going out, and so on.

Rich Picture Drawing Guidelines

For the purposes of this module, you should use the following components (all components are available online, in Word or Powerpoint or similar):

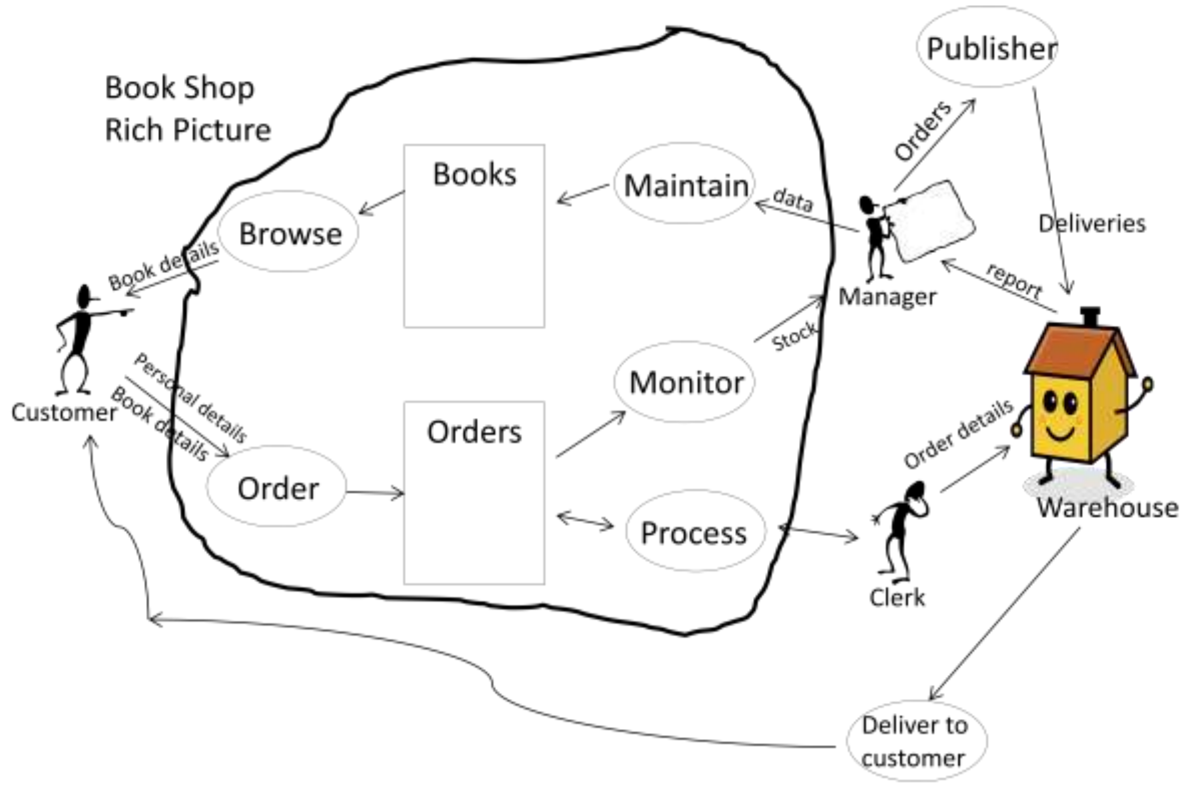
Rich Picture Components	**Comments**
Actors (with descriptive labels) Manager Clerk	Actors are the users of your system. An actor may also represent a group of users; e.g., one manager plus five data clerks will still show two actors. An actor may carry out any number of operations. Represented graphically as matchstick people.
Operations (also known as processes or functions) 	Operations specify what the system does. Each operation is executed either by an actor or another operation. Represented graphically as circles or ovals, with a descriptive label inside.
Data stores (also known as tables) Clients	Data stores are essentially the tables in your database or files in the system. It is also necessary to show the type of data they contain. Only operations may read from or write to data stores. Represented graphically as rectangles.
Arrows User details Confirmation 	Arrows show the direction of data (or information) flow amongst actors, data stores and operations. Arrows may cross the system boundary (see below). Represented graphically as single-headed arrows. Descriptive labels indicate the nature of the data or information flowing.
System boundary  (usually a solid like But my also be dashed)	The system boundary identifies those operations that you are responsible for (i.e., your area of responsibility), which means that your system must carry out everything that is inside the system boundary. You can ignore what is outside. Represented graphically as a circular line. Normally, this is the last thing you should add to your rich picture.

Note that a rich picture may represent a wider business perspective than the specific problem that you want to solve. In other words, it may contain things that you will not be required to implement. So, having drawn a rich picture, it is vitally important to define your "area of responsibility".

Here is an approach to drawing business-related rich pictures:

1. Identify the actors in the problem domain
2. For each actor, identify the operations they need to perform
3. Identify the data requirements of each operation, noting
    a. Where data will be held; and
    b. The direction of data flow between actors, operations, and stores
4. Draw the system boundary to define your area of responsibility.

Example



Remember that a rich picture without the system boundary shows the complete business model (i.e., how the whole business runs). This provides a high-level representation of the entire problem domain.

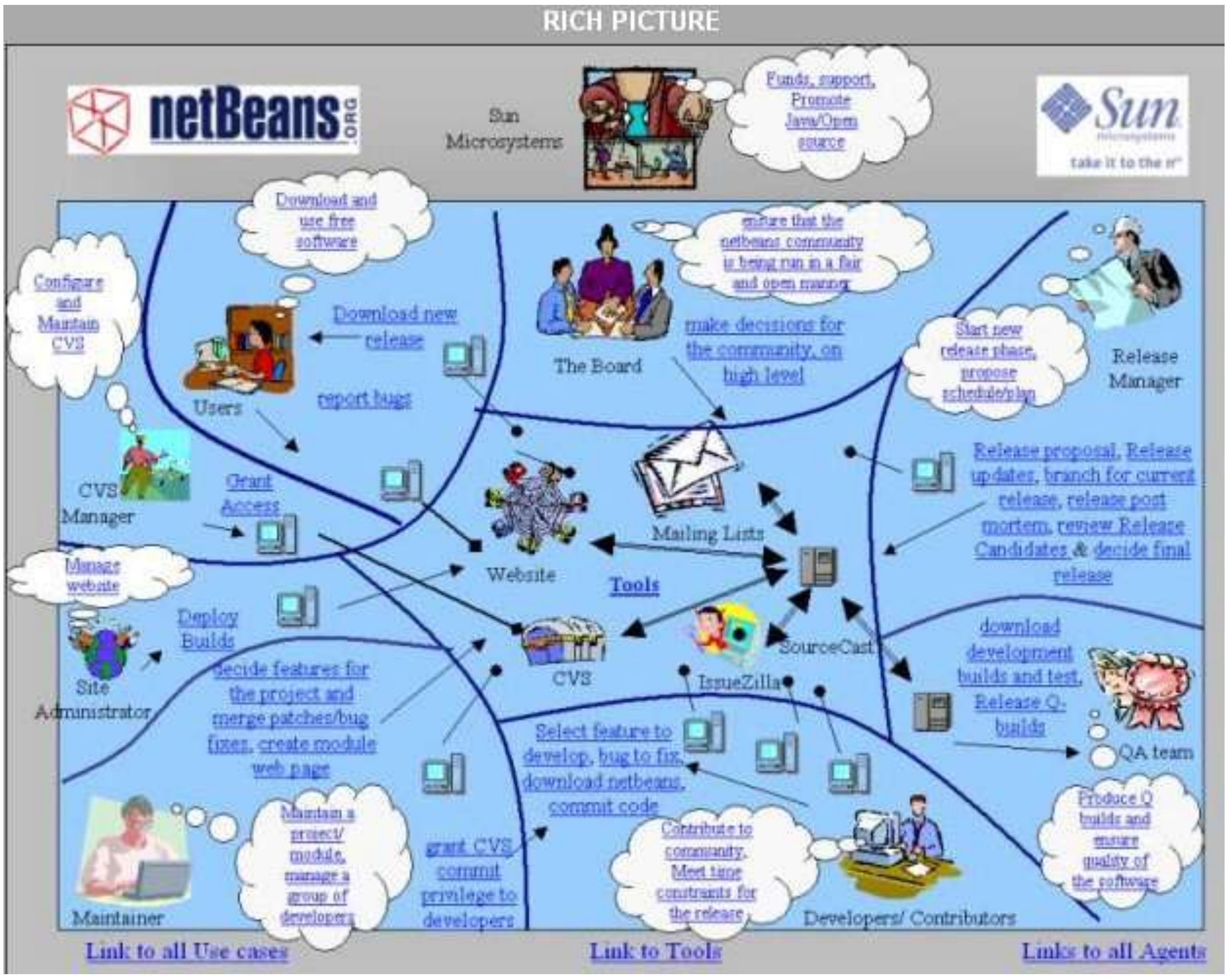
The software system you develop will rarely support the whole business. It would be too complex. By drawing the system boundary, you are defining your area of responsibility. In other words, your software will need to support only what lies within the boundary. What lies outside is someone else's responsibility.

For example, the rich picture above says that the software must enable customers to browse for books and place orders. The clerk processes those orders and arranges delivery to customers. The rich picture also says that your system is not responsible for how the books are actually delivered to the customers. Also note that how the manager communicates with the publishers is not your responsibility.

Only after drawing the system boundary can you start to think about functionality.

Example

Did you ever wonder how an organisation like Sun Microsystems operates? Here is an answer.


RICH PICTURE
netBeans.ORG
Sun Microsystems
Funds, support, Promote Java/Open source
Sun microsystems take it to the n
Download and use free software
Configure and Maintain CVS
Download new release
Users
report bugs
The Board
ensure that the netbeans community is being run in a fair and open manner
make decisions for the community, on high level
Start new release phase, propose schedule/plan
Release Manager
CVS Manager
Grant Access
Website
Mailing Lists
Tools
Release proposal, Release updates, branch for current release, release post mortem, review Release Candidates & decide final release
Manage website
Deploy Builds
Site Administrator
decide features for the project and merge patches/bug fixes, create module web page
CVS
SourceCast
IssueZilla
download development builds and test, Release Q-builds
QA team
Select feature to develop, bug to fix, download netbeans, commit code
Maintainer
Maintain a project/ module, manage a group of developers
grant CVS commit privilege to developers
Contribute to community, Meet time constraints for the release
Developers/ Contributors
Produce Q builds and ensure quality of the software
Link to all Use cases
Link to Tools
Links to all Agents

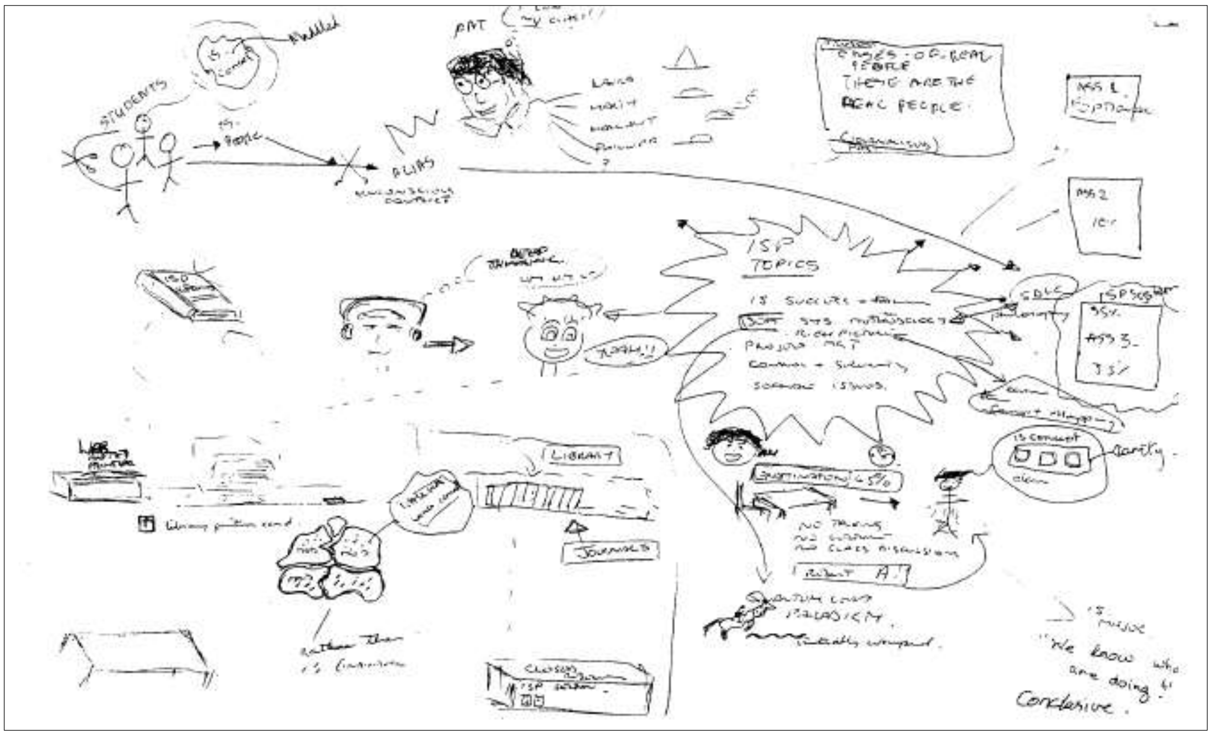
Example

To see a large rich picture with a system boundary, visit

http://onemind.wetpaint.com/page/Example+Rich+Picture

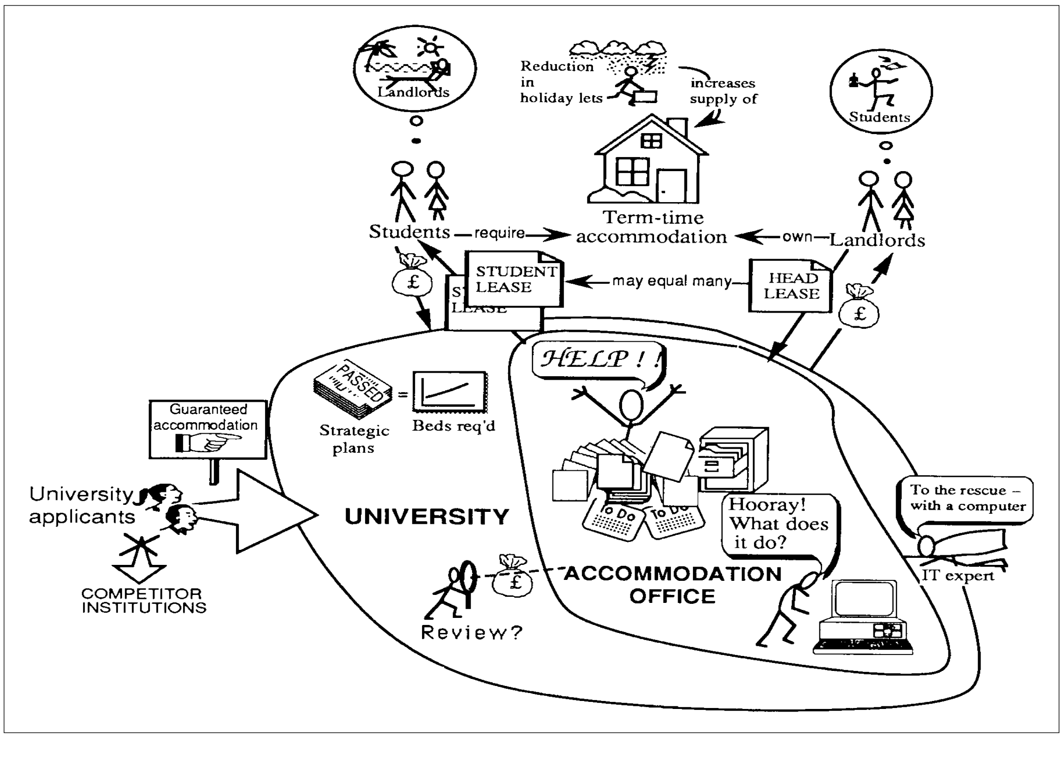
Example

Here is a very informal one, on information processing, allegedly drawn by two students in Australia



Example

What is this rich picture teling you?



Exercise 1: Draw a rich picture for the *DMU library*.

Exercise 2: Draw a rich picture for *your project scenario*.

Your first rich picture may well be incomplete the first time; so, be prepared to edit and develop it further, as your analysis and understanding of the problem deepens.