Reading Technique for Structured Analysis

Use the procedure below to perform a structured analysis of the new system by creating a data flow diagram. As you work, use the questions provided to identify faults in the requirements:

1) Read through the requirements once, and then create a high-level data flow diagram. The purpose of this diagram is to clarify the system's relationship with the outside world. Identify the data sources, destinations, and stores. Then describe the processing and specify the data flows.

    a) *Data sources and destinations* are anything outside the system that originate new data or receive finished data. A source or destination could be, for example, an external system that sends or receives messages, or a department or individual that will use the system to be built. Destinations and sources are represented by closed boxes on the data flow diagram. Label as many destinations or sources as necessary on Form A.
    b) *Data stores* represent any data or documents that are stored for any length of time. This might be, for example, a pile of papers making up a file, a database on CD-ROM, or any kind of file which can be read from and written to. Data stores are represented as boxes with one side missing. Show all of the external data stores by labeling the data store symbols on Form A.
    c) Boxes with rounded corners are used to represent *processes* or activities. Again, no distinction is made between activities which are carried out by the computer and those which are performed by some other physical device, as long as the processing is part of the system. Therefore a process symbol could refer to a person undertaking a process, a computer automatically performing some data transformation, a mechanical automated process, or any other method which achieves a required result or set of results. In the high-level data flow diagram there is only one process node which summarizes the entire process. Label the process node on Form A with a description that describes the purpose of the system described by the requirements. You will use this process node as the input to step 2.
    d) Finally, the arrows in the diagram represent *data flows*. These arrows show the flow of data from a source or data store to a process in which it is used, from process to process, or from a process to a destination or data store. Label these arrows with the actual data which is provided to the system by each of the sources and stores, or which the system sends to the destinations and stores. As always, there are no distinctions due to the form of the data.

    Q1.1 From the general requirements or your knowledge of the domain, have all of the necessary data sources, destinations, and stores been defined?
    Q1.2 Can you identify what type of data is provided by or sent to each source, destination, or store? Can the data types be defined (e.g. are the required precision and units specified for the data)?
    Q1.3 Do the requirements specify the correct source, destination, or store for each of the data flows?

2) Iteratively deepen the current processing node by constructing more detailed data flow diagrams.
    a) If there is insufficient detail to construct a lower-level diagram, or if there is only one data transformation or processing step that takes place, this process does not need to have a more detailed diagram created for it.
    b) Otherwise, use Form B to construct a lower-level data flow diagram for the current processing node. First, record the name of the current node on the appropriate line in

Form B. Then decompose the process into a series of smaller steps. Enter these processing nodes on Form B. You should divide the processing up amongst the new processing nodes in such a way that:

i) Each subproblem is at the same level of detail as the other subproblems in this data flow diagram.
ii) Each subproblem can be solved as an individual problem.
iii) The solutions to the subproblems can be combined to solve the original problem.

c) Connect the processing nodes with data flow arrows to show the flow of data between the nodes. Label each arrow with the data which is being passed along.
d) The data sources, destinations, and stores from the parent node will be needed for this lower-level diagram. Record and label them on Form B (using the appropriate symbol) and draw data flow arrows to connect them to the appropriate processes. Remember to label the arrows with the appropriate form of the data.
e) Select each of the new processing nodes you created in turn and use it as the input for repeating step 2.

Q2.1 Is all of the necessary information available to create the diagram?
Q2.2 Are the requirements clear and correct about which data flows are associated with each process? (That is, is there data flowing into a process which is not needed for the process? Is a data flow missing that is necessary for the process? Is data flowing out of a process that could not have been produced by that process?)
Q2.3 Does the flow of data occur in a logical and correct manner? Can the input to each process be recognized as the output of another process or a data source or store?
Q2.4 Is a single unique process, data source, or store described as the source of each data flow?

3) When you have expanded all of the processing nodes to an appropriate level of detail, examine the completed data flow diagram and answer the following questions:
Q3.1 Does each data flow arrow have a name associated with it? Is the data it represents defined at an appropriate level of detail in the requirements?
Q3.2 Have all of the data sources, stores, and destinations listed on Form A been used in at least one of the lower-level diagrams?
Q3.3 Can you trace through the data flow diagram using actual cases and achieve the expected result?
Q3.4 Does the flow of data within each of the lower-level diagrams make sense from what you know about the application or from what is specified in the general description?

Acknowledgements
These guidelines owe much to the following works:

Foad, K. R. Cardiff ComSc Documentation Project 1996/97. http://www.cm.cf.ac.uk/User/K.R.Foad/project/index.html.
Mynatt, Barbee Teasley. (1990) *Software Engineering with Student Project Guidance*. Prentice-Hall.
Yourdon, E., and L. Constantine. (1979) *Structured Design*. Prentice-Hall.