ICARUS
ESPRIT project 2537
Task IndusCas

A tasteful variety of specification case studies

FUNDP - Namur (B)
January 1989

I. Information Systems

1 Birthday management

(SOURCE: Spivay J.M., "An introduction to Z and formal specifications", Programming Research Group, Oxford University, August 1987)

The information to be manipulated are names and birthday dates. Each name is associated with a single date. A same date can be associated with different names. Distinct persons have distinct names. Dates are assumed to be correct The operations to be performed on this information are the following:

1. Add a new name and associate a date with it;
2. Find the date associated with a given name;
3. Find the names associated with a given date.

2 Oil terminal control system

(SOURCE: Wordsworth J.B., "Oil terminal control system - an exercise for readers of Z", IBM United Kingdom Laboratories Ltd, 1987)

An oil terminal has a number of berths at which tankers can discharge their cargoes.

When an approaching tanker asks for permission to dock, the controller will ask the system to allocate a berth for it to use. If no berth is free, the system will tell the controller so, and the tanker will be queued in the approach to the terminal. The system assumes that there will be enough room for any number of waiting tankers.

On docking, a tanker occupies the allocated berth, unloads its cargo, and so on. When it is ready to leave, the controller will notify the system so that the berth is available for reuse and the tanker is deleted from the system. A tanker's leaving a berth might mean that a queueing tanker can come and occupy it. The system will identify the tanker at the head of the queue to the controller, and allocate that berth to the tanker.

The system has enquiry facilities so that the controller can get information about which tankers are queueing, which berths are occupied and by which tankers, and which berths are free.

3 Airline reservation

(SOURCE: van Lamsweerde A., 1983)

An interactive airline reservation system receives a command flow according the following syntax:

$$
\begin{array}{rcl}
< command - flow > & ::= & < commands > END \\
< commands > & ::= & < commands >< command > \\
< command > & ::= & FLIGHT < number >< number > \\
& & \mid GROUND < number > \\
& & \mid SEAT < number >< string >< string > \\
& & \mid CANCEL < number >< string >< string > \\
& & \mid PRINT < number >
\end{array}
$$

$< number >$ denotes a positive 3-digit integer.

$< string >$ denotes any string with a maximum length of 30 characters.

Words in a command may be separated by any number of blank characters.

Each input must produce either an error message if the command is not allowed, or a message expressing that the command has been processed.

Commands have the following semantics:

$FLIGHT < number1 >< number2 >$:
: Flight number $< number1 >$ with a capacity of $< number2 >$ passengers is added to the system. An error message is issued if that flight already exists.

$GROUND < number1 >$:
: Flight number $< number1 >$ is removed from the system. All reservations for that flight, if any, are also removed. An error message is issued if the flight doesn't exist.

$SEAT < number1 >< string1 >< string2 >$:
: Books a seat in flight number $< number1 >$ for a passenger whose name and first name are $< string1 >$ and $< string2 >$, respectively. Two passengers may have a same name; in such a case, their first name must be distinct from one another. Appropriate error messages are issued if flight number $< number1 >$ doesn't exist or if the constraint on names is violated. Otherwise, a free seat number is returned, if there is one, or an appropriate message is issued if the flight is completely booked.

$CANCEL < number1 >< string1 >< string2 >$:
Cancels the reservation on flight number $< number1 >$ for the passenger whose name is $< string1 >$ and first name is $< string2 >$. Appropriate error messages are issued if the flight or the reservation do not exist.

$PRINT < number1 >$:
Prints a passenger list, alphabetically sorted, for flight number $< number1 >$. An error message is issued if the flight doesn't exist.

END : Ends the interactive session.

4 Library system

(SOURCE: Kemmerer R.A.,"Problem set for the 4th International Workshop on Software Specification and Design", IEEE Catalog Nr:TH0181-8, 1987, pp. x)

Consider a small library database with the following transactions:

1. Check out a copy of a book / Return a copy of a book;
2. Add a copy of a book to / Remove a copy of a book of the library;
3. Get the list of books by a particular author or in a particular subject area;
4. Find out the list of books currently checked out by a particular borrower;
5. Find out what borrower last checked out a particular copy of a book.

There are two types of users: staff users and ordinary borrowers. Transactions 1, 2, 4 and 5 are restricted to staff users, except that ordinary borrowers can perform transaction 4 to find out the list of books currently borrowed by themselves.

The data base must also satisfy the following constraints:

1. All copies in the library must be available for checkout or be checkout;
2. No copy of the book may be both available and checkout at the same time;
3. A borrower may not have more than a predefined number of books at one time.

5 Flying-club management

SOURCE: Anna Gram, "Raisonner pour programmer", Dunod Informatique, 1986.

The flying-club system must meet the following general needs:

- **Pilot management**: Each pilot has its own characteristics (address, aeronautical qualifications,...) and a "club account". This account is charged when the pilot performs a paying flight or when he buys products the club supplies.

- **Plane management**: The system must manage a log-book describing the use of planes. Plane management must take into account plane buying and selling.

Inputs of the system:

- **Concerning pilots**: The system receives the following at each new registration: pilot's full name, address, license number and further qualifications.
- **Concerning planes**: The system receives the following each time a plane is bought by the club: plane registration number, type, manufacturing year and trip-recorder. The information concerning a plane also includes a per hourly hiring rate which must sometimes be updated.
- **Concerning flights**: After each flight, the system receives the pilot's name, the plane identification, the date and time of take-off and landing, and whom the flight must be invoiced to (if it is not the pilot himself). A flight registration must be charged to the corresponding account in the system.
- **Concerning accounts**: Accounts are by debited (see above) or credited by deposits. Each account operation must specify the date and amount.

The system must be able to produce the following outputs:

- **About pilots**:
  - the list of the name of the pilots being currently involved in the club;
  - the characteristics of a given pilot;
  - for each pilot, the list of flights performed by him between two given dates and the list of changes occurred onto his account between two given dates.
- **About planes**:
  - the list of flights performed so far, sorted by plane number and, for a given plane, by date;
  - the list of flights between to given dates for a given plane;
  - the characteristics of a given plane;
  - the total flying time of a given plane;
  - the hourly hiring rate for every plane;
  - the list of registration number of the planes.
- **About accounts**:
  - the list of changes occurred on the accounts between two given dates.

6 "CAVIAR"

(SOURCE: Flinn B. and Sørensen I.H., "CAVIAR: A case study in specification" in Hayes I., "Specification case studies", Prentice Hall International, 1987)

This specification of a **C**omputer **A**ided **V**isitor **I**nformation **A**nd **R**etrievial system resulted from the analysis of a manual system concerned with the recording and retrieval of data about arrangements for visitors and meetings at a large industrial site.

Visitors come to the site to attend *meetings* and/or consult company employees. A visitor may require a *hotel reservation* and/or *transport reservation*. Each meeting is also required to take place in a designated *conference room*, at a certain *time* for a certain *period*. A meeting may require the use of a *dining room* for lunch, on a particular date. Booking a dining room requires *lunch information* including the number of seats required. Each conference room booking requires *session information* about resources required for use in the meeting, e.g., viewgraphs, projectors.

The main operations required can briefly be described as facilities for booking, changing and cancelling the use of all resources involved.

7 Inventory system

(SOURCE: Bubenko J.A., "On concepts and strategies for requirements and information analysis" in "Information modeling", J.A. Bubenko (ed.), Chartwell-Brat, 1983, pp. 125-169)

Assume a retail company which has set up the following operations to support its regular activities:

- Get the current quantity in inventory of a particular type of article. This request may occur at any time (T); the response should reflect the inventory [1] situation no longer than ΔT units ago.
- Get the yesterday's sales (in terms of dollar value) of a particular group of items. The request may occur at the beginning of each day (D); the response should correctly reflect the situation for day $D-1$.
- Get the total sales for last month for a particular type of item. This request may occur at the beginning of each month (M); the response should correctly reflect the situation for month $M-1$.
- Record the sales for a particular type of item.
- Record the delivery from vendor for a particular type of item.
- Record the unit price of a particular type of item.
- Record which group a type of item belongs to.
- Introduce a new type of item.
- Cancel a particular type of item.

[1] Note: the confusion between two distinct entities under the same name viz. the "physical" inventory and the computer representation of it.

8 Stock management

(SOURCE: van Lamsweerde A., 1983)

A company is composed by about 500 supermarkets and a big central warehouse. About 4000 different items are stored in the warehouse (Two different quantities of a same product are taken as distinct items). The items are coming from about 400 suppliers. Each item is coming from exactly one supplier.

Items are organized into 80 product groups (i.e. tinned fruits, deep-frozen vegetables, wine and liquor,...). The biggest product group contains about 120 items. Each item belongs to exactly one product group.

The warehouse gets from suppliers items that have been ordered by warehouse managers. These items are then distributed to supermarkets according to their orders.

The price of an item is the same whichever supermarket ordered it (the price includes transportation charges; no discount for big quantity is granted).

The computer system is in charge of stock management and must be able to perform sales analysis.

Every day the orders coming from the supermarket and the deliveryslips coming from suppliers until 4 p.m.; they are then processed by batch (documents arriving past 4 p.m. are processed the next day).

The functionalities of this batch processing system are the following:

1. The stock inventory is updated from the batch of deliveries.
2. Supermarket orders are processed as follows: if, for a given item, the stock level is sufficient for all orders, the orders are fully granted. But if it is not sufficient, the remaining quantity is parcelled out between the orders proportionally to the quantities desired. So, an order may be delayed with respect to some quantity of various items.
3. Delayed orders must be processed before new ones.
4. Order processing results in updating the stock inventory according to the quantities of items that have been effectively supplied to supermarkets.
5. A record is kept of the current state of each order.
6. For each order that has been partially or fully granted, an invoice is generated. The invoice includes the identification of the target supermarket, the delivered quantity and price of each item ordered, and the total amount due.
7. After processing all orders, the system checks the stock inventory. For each item, if the quantity in the warehouse is less than 100 units, an order is sent to the corresponding suppliers for 1000 units of that item.
8. An order to a suppliers contain the identification of the supplier, the identification, the quantity and price of the items to be ordered, and the total amount due.

[0]Note: To what extend are items (4) and (5) above functional requirements or implementor concerns.

Every week, the system must produce a sales analysis. For each product group, the system generates the following report:

SALES ANALYSIS	
Product group: $<\mathit{pgroup}>$	Date: $<\mathit{date}>$
Total amount of delayed orders at previous sale analysis	\$ $<\mathit{number}>$
Total amount of orders received this week from supermarkets	\$ $<\mathit{number}>$
Total amount of delayed orders processed this week	\$ $<\mathit{number}>$
Total amount of non-delayed orders processed this week	\$ $<\mathit{number}>$
Total amount of orders currently delayed	\$ $<\mathit{number}>$

9 Conference organization

(SOURCE: Olle T.W. et al., "Information system design methodologies: a comparative review", A.A. Verrijn-Stuart (eds), North-Holland, 1982)

The information system which is to be designed is that necessary to support the activities of both a program committee and an organizing committee involved in arranging an IFIP working conference. The involvement of the two committees is seen as analogous to two organizational entities within a corporate structure using some common information.

The following activities of the committees should be supported:

- **Program Commitee**:
  1. Preparing a list to whom the call for papers is to be sent.
  2. Registering the letters of intent received in response to the call.
  3. Registering the contributed papers on receipt.
  4. Distributing the papers among those undertaking the refereeing.
  5. Collecting the referees' reports and selecting the papers for inclusion in the program.
  6. Grouping selected papers into sessions for presentation and selecting chairman for each session.
- **Organizing comittee**:
  1. Preparing a list of people to invite to the conference.
  2. Issuing priority invitations to national representatives, working group members and members of associated working groups.
  3. Ensuring all authors of each selected paper receive an invitation.
  4. Ensuring authors of rejected papers receive an invitation.
  5. Avoiding sending duplicate invitations to any individual.
  6. Registering acceptance of invitations.
  7. Generating a final list of attendees.

II. Real time

10 Heating system (simple version)

(SOURCE: Genesereth M.R., Nilson N.J., "Logical foundation for artificiel intelligence", Morgan Kaufmann Publishers, 1987)

We consider a system for a thermostat controlling a furnace. The thermostat's state includes a temperature indicator and a state sensor.

- The temperature indicator is either *down*, *up* or *ok*. It is *down* whenever the ambient temperature is more than 5 degrees cooler than a predefined setting. It is *up* whenever the ambient temperature is more than 5 degrees warmer than the setting temperature. Otherwise, it is *ok*.
- The state sensor is either *on* or *off*. It is *on* when the furnace is activated and remains *on* until deactivation. It is *off* when the furnace is deactivated and remains *off* until activation.

The thermostat has different actions: it can turn on the furnace, turn off the furnace, or do nothing. Actions are governed by the following rules :

- The thermostat turns on the furnace whenever the ambient temperature falls below the -5 degrees limit or whenever there is a request expressed by the external end-user.
- It turns off the furnace whenever the ambient temperature reaches the +5 degrees limit or whenever there is a request expressed by the external end-user.
- Otherwise, it does nothing.

11 Telephone network (simple version)

(SOURCE: Morgan C., "Telephone network" in Hayes I., "Specification case studies", Prentice Hall International, Englewood Cliff, N.J., 1987, pp.73-87)

Consider a very simple telephone network in which connections may be established between pairs of telephones. A request may be made for the connection of a given phone to any other; if the request cannot be satisfied immediately, it will be stored by the network and satisfied if possible at some time later (for example, the other phone might be engaged).

Only three network operations are to be considered; they are

Call : request a connection between two telephones.

Hangup : terminate a connection.

Engaged : indicate whether a given telephone is currently engaged, and if so, with which phone.

12 Patient monitoring system (simple version)

(SOURCE: Stevens W.P., Myers G.J. and Constantine L.L., "Structured design", IBM Systems Journal 13 (2), 1974, pp. 115-139)

A patient-monitoring program is required for a hospital. Each patient is monitored by an analog device which measures factors such as pulse, temperature, blood pressure, and skin resistance. The program reads these factors on a periodic basis (specified for each patient) and stores these factors in a data base. For each patient, safe ranges for each factor are specified (e.g., patient X's valid temperature range is 98 to 99.5 degrees Fahrenheit). If a factor falls outside of a patient's safe range, or if an analog device fails, the nurse's station is notified.

13 Heating system (more complex version)

(SOURCE: White S.,"Problem set for the 4th International Workshop on Software Specification and Design", IEEE Catalog Nr:TH0181-8, 1987, pp. ix)

The controller of an oil hot water home heating system regulates in-flow of heat, by turning the furnace on and off, and monitors the status of combustion and fuel flow of the furnace system, provided the master switch is set to HEAT position. The controller activates the furnace whenever the home temperature, t, falls below $t_r - 2$ degrees, where t_r is the desired temperature set by the user. The activation procedure is as follows:

1. The controller signals the motor to be activated;
2. The controller monitors the motor speed and once the speed is adequate it signals the ignition and oil valve to be activated.
3. The controller monitors the water temperature and once the temperature is reached a predefined value it signals the circulation valve to be opened. The heated water then starts to circulate through the house.
4. A fuel flow indicator and an optical combustion sensor signal the controller if abnormalities occur. In this case the controller signals the system to be shut off.
5. Once the home temperature reaches $t_r + 2$ degrees, the controller deactivates the furnace by first closing the oil valve and then, after 5 seconds, stopping the motor.

In addition the system is subject to the following constraints:

1. minimum time for furnace restart after prior operation is 5 minutes.
2. furnace turn-off shall be indicated within 5 seconds of master switch shut off or fuel flow shut off.

14 Post office

(SOURCE: Yonezawa A., "Specification and verification techniques for parallel programs based on message passing semantics", Ph.D. Thesis, M.I.T., Techn. report MIT/LCS/TR-191, pp. 181-199)

A number of customers and mail collectors visit the post office, possibly simultaneously. The post office has only one door for customers and collectors. Inside the post office, there is a counter section which has several counters and a mail box corner which has a mail box.

After a customer enters the post office through the door, if he needs stamps, he goes to the counter section, otherwise he goes to the mail box corner. At the counter section, a customer gets the stamps he needs and then, if he is carrying letters, he goes to the mail box corner, otherwise he goes out of the post office through the door.

Customers are served at the counter section on a first-come-first-served basis, but the time spent at the counter varies from person to person. At the mail box corner, a customer puts all the letters he has been carrying in the mail box and goes out through the door.

A collector also enters the post office through the door and then goes to the mail box corner. At the mail box corner, the collector collects all the mail in the mail box after waiting in the queue, if there is one, and then he carries the collected mail out of the post office through the door. Customers and collectors make a single queue at the mail box corner and arrive and leave the corner on first-in-first-out basis.

Components of the post office, collectors and customers have their own local time. Thus, arrivals of customers and collectors at these components are in general mutually independent. Furthermore, we assume that the walking speed of customers and collectors may vary from person to person. So, for example, a customer arriving at the door after another customer may arrive at the counter section before him. Services at different counters are carried out concurrently, and of course depositing and collecting the mail in the mail box corner take place independently of the activities at the counter section.

15 Journal editing process

(SOURCE: Zisman M.D., "Pattern-directed inference system", ed. Waterman D.A., Hayes-Roth F., Academic Press, 1978, pp.66)

A program for automated support to the editing process is required for a journal. The waiting states in which the system can exist are the following:

1. waiting for a paper to arrive;
2. waiting for the editor to designate referees;
3. waiting for the referee to respond;
4. waiting for the referee to submit his report;
5. waiting for all reports to arrive;
6. waiting for the editor to make a decision.

Next we describe what should happen in each of these states:

1. When a paper arrives, an acknowledgment letter to the author must be generated and a request must be issued to the editor asking him to designate referees.

2. If the editor does not respond within two weeks, a reminder letter must be generated to him and so on every two weeks. When the response is obtained letters to each referee must be generated to request their service.

3. If a referee does not respond within two weeks, a reminder letter, must be generated to him. If he responds and states that he cannot review the paper, the editor must be informed and a request for designating another referee must be issued. If the referee agrees to review the paper, a one-month period must be allocated to him for submitting his report.

4. If the referee does not respond within one month, a reminder letter must be generated. When the report is received, the editor must be informed of its arrival.

5. When all reports are received, a request must be issued to the editor asking him to make a decision.

6. If the editor does not make a decision within two weeks after all reports are received, generate a reminder letter must be generated. When a decision is made, the author and executive editor must be informed.

16 Patient monitoring system (more complex version)

(SOURCE: Dubois E.,1989)

The Westminster Hospital in London has a cardiac surgery unit where open heart operations are performed on patients using the technique of profound hypothermia. After such a procedure there is considerable danger to the patient as his body readjusts to normal temperature control. In particular, post-operative patients have a large excess of body fluid. During the period of adjustment it is essential that the patient's body functions and vital signs be carefully monitored and, where possible, adjustments be made in time to preserve life.

The hospital has installed in its post-operative intensive care unit an on-line computer system which is used to monitor patients' life functions, such as blood pressure and heart rate, record data concerning the patients in a database, and raise an alarm when any of the monitored parameters lies outside acceptable critical limits. The case study is concerned with the software for controlling this system. However, it is not a simple copy of the system. Rather, it is an exercise in understanding the nature of the problem and then engineering a design that be suitable for a re-implementation of the system.

The intensive care unit in this study has sixteen beds. Each bed is equipped with a visual display computer terminal (VDU) and a set of monitoring sensors that can be attached to a patient. There is also a central monitoring station where a VDU is provided for use by hospital staff. The system is controlled by a single processor with disc storage and a magnetic tape unit.

All interactions with the system are by hospital doctors and nurses and there are no special purpose computer staff employed, except for on-call maintenance engineers. The system is thus embedded within the normal functioning of the cardiac unit and has the real-time response problems of interacting with the automatic monitors.

When a new patient is admitted to the unit a member of the medical staff is responsible for activating the monitoring system and initializing the patient's data. The first step is to prompt a bedside VDU. This causes the system to respond with instructions which are followed by the hospital staff. Initially, the system prompts on the VDU for the input data such as the patient's identity, the initial values of parameters that are not sensed directly and the acceptable upper and lower limits that are to be monitored on each parameter. After this phase, the sensors are connected by a nurse or doctor, the system responding after each connection to indicate either that the function is correct or that the sensor should be adjusted.

Once the patient has been successfully connected to the system the terminal displays a menu of actions available to hospital staff. This menu includes possible data inputs and enquiries about the patient's condition. On-line monitoring continues until the patient is disconnected.

The monitoring data on patients is collected in two ways. Firstly, the sensors attached to each patient detect values such as blood-pressure and temperature. These are polled at regular intervals and their readings recorded. Secondly, the VDU by each patient's bed is used for manual input of data by hospital staff. This data relates mainly to fluid inputs and outputs. From the measures given, the system calculates the fluid balance for the patient. Manual input can also be used to replace expected data from a failed sensor, an essential safety requirement. The upper section of the VDU screen is used to display the latest monitored readings and the last item of manually input data. The system raises an alarm either when no reading is obtained, due to a fault in the apparatus, or when readings show that the patient's vital signs have moved beyond the acceptable limits.

If the values detected by the system fall outside the limits set for a particular patient, then the system will activate a light over the patient's bed and a buzzer and light in a central monitoring booth. Information on the nature of the particular emergency will be displayed on the bedside VDU. These alarms will continue to function until a member of the medical staff types a code at the patient's VDU. During the alarm the system will still continue to record the patient's data for subsequent analysis.

In addition to the on-line monitoring, the system supports other functions. Hospital staff can use a bedside VDU to request that the system produce analyses of the historical data held on a patient. This can be displayed in a graphical form either on a VDU or using a hard copy printer. A doctor can note the drugs that a patient is to receive, and this information can be used to prompt the nursing staff who administer the drugs. Doctors can also alter the acceptable limits for the factors associated with a patient. There is one central VDU that is located at the nurse's monitoring station. This can be used to obtain information on any patient and will also display warning messages concerned with any patient. The system also stores historical data on each patient which is used to produce regular report summaries.

When a patient ends his stay in the unit, either by being discharged back to the ward or by dying, a number of the hospital staff will disconnect his sensors and type any final data into the system using the bedside VDU. This patient's data will remain available on-line for a further 48 hours, after which time it is archived onto tape and responsibility for it passed to the hospital's central data processing unit. Again, the system will prompt the actions that have to be taken at this stage.

17 Railway

(SOURCE: "The problem set of the Workshop on the Analysis of Concurrent Systems", in Danvir B.T., Harwood W.T., Jackson M.I., Wray M.J., "The analysis of concurrent systems", L.N.C.S. 207, 1985, pp. 117)

The problem is to produce a formal specification of a railway system as described below.

A railway system is made from lengths of a single track joined at crossing points (a length of track may consist of one or more sections). A train may leave a crossing point on any unoccupied section of track connected to it. The system is to be run so that no section of track can have more than one train on it at any time. The spacification should be independent of the number of trains in the system.

18 Lift system

(SOURCE: Davis N., "Problem set for the 4th International Workshop on Software Specification and Design", IEEE Catalog Nr:TH0181-8, 1987, pp. x)

An n lift system is to be installed in a building with m floors. The lifts and the control mechanism are supplied by the manufacturer. The internal mechanisms of these are assumed (given). The problem concerns the logic to move lifts between floors according to the following constraints:

1. Each lift has a set of buttons, one for each floor. These illuminate when pressed and cause the lift to visit the corresponding floor. The illumination is cancelled when the corresponding floor is visited by the lift.
2. Each floor has two buttons (except ground and top floors), one to request an up-lift and one to request down-lift. These buttons illuminate when pressed. The illumination is cancelled when a lift visits the floor and is either moving in the desired direction, or has no outstanding requests. In the later case, if both floor request buttons are pressed, only one should be cancelled. The algorithm to decide which to service first should minimize the waiting time for both requests.
3. When a lift has no request to service, it should remain at its final destination with its doors closed and await further requests *(or model a 'holding' floor).*
4. All requests for lifts from floors must be serviced eventually, with all floors given equal priority *(can this be proved or demonstrated?).*
5. All requests for floors within lifts must be serviced eventually, with floors being serviced sequentially in the direction of travel *(can this be proved or demonstrated?).*
6. Each lift has an emergency button which when pressed causes a warning signal to be sent to the site manager. The lift is then deemed 'out of service'. Each lift has a mechanism to cancel its 'out of service' status.

19 ABRACADABRA protocol

(SOURCE: Oregas F.,1989)

The ABRACADABRA protocol is a rather simple protocol for linking two communicating stations. This description was used by CCITT.

Many people have studied the alternating bit protocol which models a unidirectional flow of information with a positive handshake on each transfer. This protocol is too simple to represent a number of complexities found in real communication protocols.

The purpose of this example is to describe a more realistic protocol which has Alternating Bit sequence number, retransmission on timeout, acknowledgements, connection and disconnection.

19.1 Prose description

The protocol operates over a full duplex, short delay, transparent communications medium between two stations. The medium may occasionally loose messages, but will not misorder, corrupt, invent or duplicate messages.

The two stations communicate by transfer Protocol Data Unit (PDUs). The communication protocol is two-way simultaneous, symmetrical and reliable. Communication proceeds in the three cyclically sequential phases: connection, data transfer and disconnection. Only the PDUs defined are possible.

19.1.1 Connection phase

The initiating station (which may be either) sends a Connection Request (CR) PDU. If it receives a Connection Confirmation (CC) PDU, it proceeds to the data transfer phase. If it receives a Disconnection Request (DR) PDU it proceeds directly to the disconnection phase and processes the DR. If it receives any other kind of PDU it enters the disconnection phase and sends a DR.

If no response is received within period P, the CR is transmitted.

A maximum of N connection attempts are permitted (i.e. N periods of value P); after these the connection attempt is abandoned and a local user within the station is informed by a disconnection indication. The responding station will reply CC on receipt of CR when no connection is set up; the connection may also be refused by DR.

19.1.2 Data transfer phase

The initiating station (which may be either) sends a Data Transfer (DT) PDU. If it receives an acknowledgement (AK) PDU it may send a further DT. If the correct response is not received within period P the DT is transmitted (i.e. N periods of value P). After this time the connection is abandoned and the local user within the station is informed by a disconnection indication.

Each DT and AK bears a one bit sequence number. Successive DTs carry the next number in sequence (i.e. the other value) starting at 0 after connection. The correct acknowledgement to a DT contains the sequence number of the next DT expected (i.e. the other value).

Data transfer and sequence numbering in both directions are independent.

Any other PDU except DR is an error, resulting in entry to the disconnection phase and transmission of DR. If DR is received then the disconnection phase is entered and the DR is processed.

19.1.3 Disconnection phase

The initiating station (which may be either) sends a Disconnection Request (DR) PDU. If it receives a Disconnection Confirmation (DC) PDU, it considers the disconnection to be complete; the same is true if DR is received instead. Any other kind of PDU is ignored.

If a DR response is not received within period P the DR is retransmitted.

A maximum of N disconnection attempts are permitted (i.e. N periods of value P). After this the disconnection attempts are presumed to be successful.

A DR may be sent at any time, for example: after CR and before CC, after DT and before AK, and when there is no connection at all.

19.2 Local user interface

Each station is presumed to support the local user interface to the communication protocol entities which are defined above. This interface closely reflects the protocol operations, except that user data units (UDU) are used instead of protocol data units (PDU).

CR,CC UDUs
DT,AK UDUs
DR,DC UDUs

19.3 Communications medium interface

The communications medium interface is presumed connectionless and is accessed through Unit Request (UR) Medium Data Unit (MDU) transfers.

19.4 Protocol model

The protocol should to be modeled as two protocol entities conversing through two local interfaces.