**The BCS Professional Examinations**
**Diploma**

**October 2006**

**EXAMINERS' REPORT**

**Systems Analysis**

## General Comments

It would be helpful to examiners when marking a large number of scripts if:

- invigilators could remember to ask students to indicate on the front of their examination paper which questions they have answered.
- A general weakness among candidates who did not attain high marks is a tendency to make a statement without an accompanying explanation; where this occurred in particular questions is highlighted below.
- As ever, the other overall weakness was a tendency to not answer the question; specific examples are indicated under individual questions below.

Question 1 is compulsory. Attempt TWO other questions.

**1.** The Chan family are the proud new owners of the Paradise Beach Guest House with its 10 en-suite rooms, bar and restaurant. They have asked you to develop a system that will manage the room bookings processes, keep details of additional services that guests purchase during their stay and produce the final bill for a stay. The Chan's describe the processing involved as follows:

- A booking or visit starts with a room request coming in from a potential guest. We take details of the reservation request and search for availability by looking in the desk-diary to see which rooms are available for the required dates. We have 2 single, 6 double and 2 family rooms that can accommodate up to 5 persons in each.

- If the required rooms are available we create a record of the reservation and include details of the room allocated to the reservation, start and end dates and the number of guests staying in the room. At this stage we also check to see if the potential guest has stayed with us before or not. If they have, we find their details, check they are still accurate and then add their details to the reservation. If they haven't stayed with us before, we take the name of the guest making the reservation and their contact details and add these to our records.

- If a guest wants to reserve more than one room we create a separate reservation record for each room reserved.

- When guests arrive at the start of their stay, we find their reservation and personal details, checking that their personal details are still correct and updating them as necessary. We then book them in for their stay, we call this an 'occupancy'. Local laws require that we record the names and age of all guests and which room they are staying in.

- We provide a restaurant, cinema and theatre booking service for our guests. This process involves little more than a telephone call to the organization and a verbal agreement being reached between the Guest House and the organization.

- During their stay at the Guest House guests may purchase or hire additional services and products from us, such as newspapers, cycles, maps, food and drinks. We need to record all of these so that we can charge for them at the end of the stay.

- At the end of their stay guests request their final bill. We would like the system to calculate and produce this final bill. We then give it to the guest so that payment can be made. We keep a record of how much each stay in a particular room was billed for so that we can calculate income per room over a given period of time.

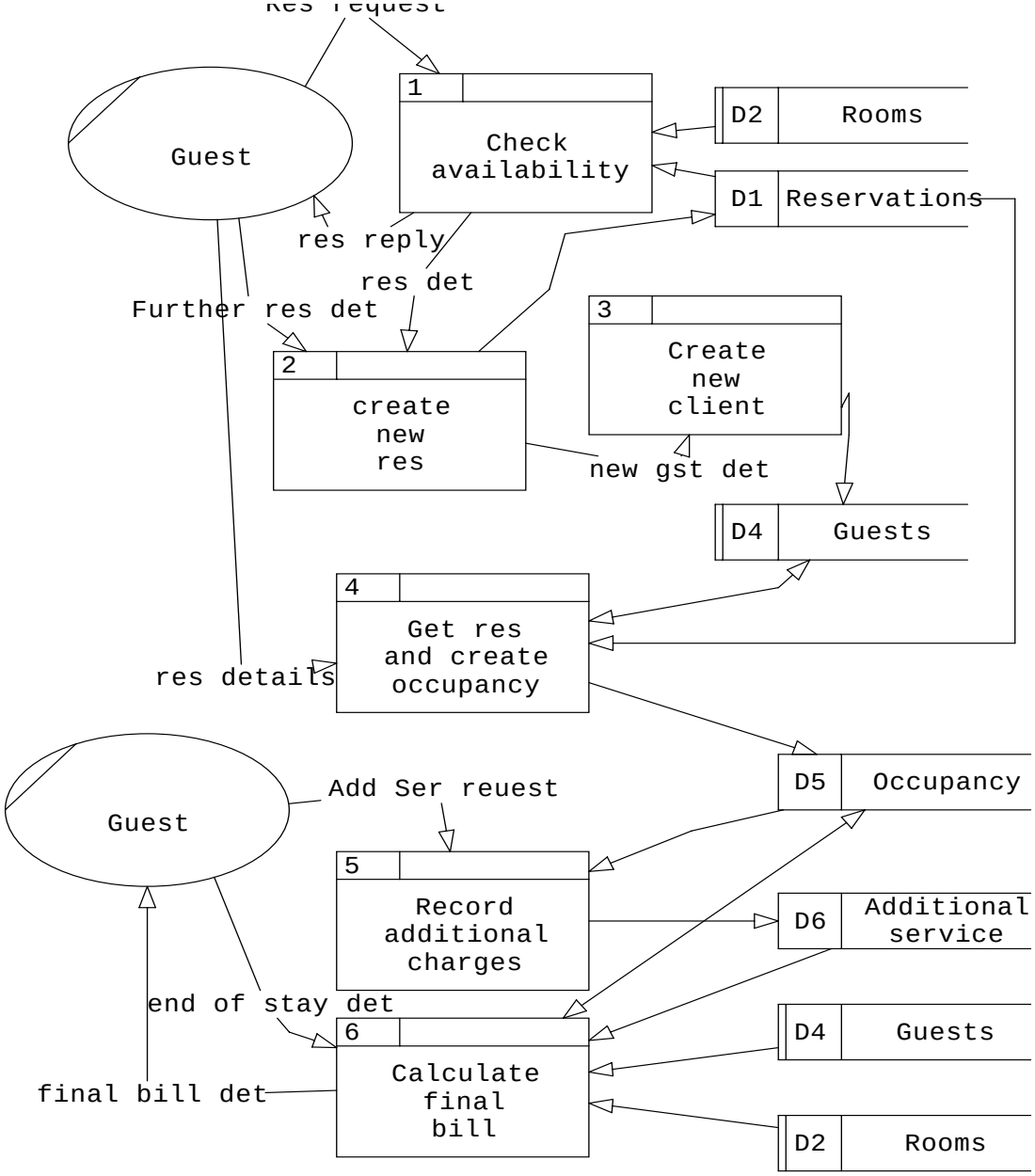
*a)* Draw a Top Level Current Logical Data Flow Diagram for the above scenario. **(20 marks)**

*b)* Produce an Entity Relationship Diagram (Logical Data Structure) and a set of normalised tables for the above scenario. **DO NOT** show evidence of the normalisation process. **(20 marks)**

*c)* Explain, **USING AN EXAMPLE** from the Data Flow Diagram you produced in response to Question 1 *a)*, the meaning of the term 'a balanced set of data flow diagrams'. **(10 marks)**

**Question 1 Answer Pointers**

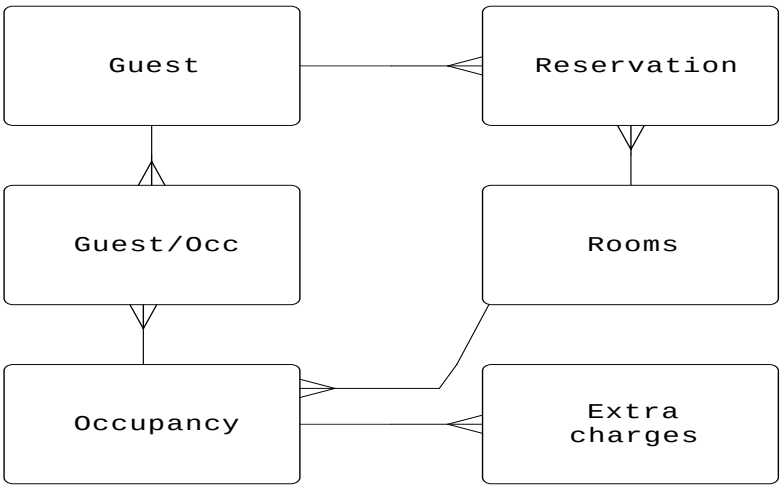
**1a) Data Flow Diagram** – Suggested solution (other notation used consistently will be accepted)




**Q1a. Marking Scheme**

| | | |
|---|---|---|
| Processes: | 1 each up to 5 | |
| Data Stores: | 1 each up to 3 | |
| Data Flows: | 0.5 each up to 4 | |
| External Entities: | 2 for guest | |
| Logical/not physical | Up to 6 marks | **Total 1a: 20 marks** |

**1b) ERD (LDS)** (other suitable notation will be accepted).




**Tables**

| | |
|---|---|
| Guest | (**guest#,** gname, gadd, gtel, gemail, gage) |
| Reservation | (**res#,** datemade, fromdate, todate, room#,*guest#)* |
| Guest/Occ | (**guest#, occ#**) |
| Rooms | (**room#,** roomtype, roomrate) |
| Occupancy | (**occ#,** startdate, enddate, finalbillamount, *room#*) |
| Extra charges | (**ec#**, ecdate, ectime, eccode, ecamount, ecreceipt#, *occ#*) |

**Notes**

1. The decomposed MM between 'Guest' and 'Occupancy' is to allow for many guests staying in one room at one time
2. Some candidates may show a relationship between 'Reservation' and 'Occupancy'. This is not unreasonable but may be more representative of a process relationship/sequence rather than a data relationship.

**1b. Marking Scheme**

| | |
|---|---|
| Entity types | 1 mark each to max of 6 |
| Relationships with appropriate degrees & membership classes | 1 mark each to max of 5 |
| Table types, consistency with LDS & appropriate PK: | 0.5 each to max 3 |
| FK postings: | 0.5 each to max 3 |
| Other attributes placed to result in normalised tables and support processing: | to max of 3 |

Total 1b: 20 marks

**1c Answer Pointers**

A set of diagrams refers to the different levels of DFD's, ie. Context, top or L1 and lower level child diagrams.

A balanced set refers to the need for all the I/O and stores at each level to remain consistent with each other, i.e., the number of I/O's at the parent level balances with the lower level and that once a store is shown at one level it must be shown at all child levels.

**1c. Marking Scheme**

| | | |
|---|---|---|
| 3 marks for explanations of "Set" and Balanced | 6 marks | |
| 4 marks for an example | 4 marks | Total 1c: 10 marks |

## Examiner's Comments

**Overall comments regarding Question 1**

Despite this being a predictable question the levels of technical awareness and skills demonstrated were lower than previous years. Indeed, the overall average mark for this question was 16.7/50. This relatively low mark depressed the marks of individual candidates across all centres. Obviously there were some high scoring candidates in Q1a and Q1b, but Q1c was almost universally answered incorrectly with most candidates scoring 0/10. Please see the detailed comments below regarding each aspect of this question and hopefully future candidates will seek to ensure that they understand the errors identified and ensure that they do not repeat them.

### Question 1a

Pleasingly the major processes, stores and external entities were identified by almost all candidates, but the most frequent errors included:

1. Not naming/labelling data flows that go from process to process.
2. Labelling (passive) data flows with verbs, i.e., 'get reservation', 'add new customer', 'check the bill', etc. Data flows are passive items of data flowing round a system, they carry no responsibility for processing or doing any actions and their names should reflect this.
3. Many candidates described their data flows with names that were a too general, for example 'info' and 'data'. Alone these words do not describe the actual data flow and often their use leads to another frequent error, duplication of flow names.
   Candidates should also seek to make process names meaningful, yet concise. Examiners know how challenging this is but it is still expected that the process name
4. suggests some processing. Process names such as 'billing', 'checking', 'reservations', 'guest handling' are not sufficiently descriptive of their processing responsibilities. The suggested solution provides examples of process names that overcome this problem.
5. Many candidates continue to include processes that have input but no output, or output but no input, or in some cases no input or output at all. The basic rules to remember is that
   a) process is expected to have at least one input (to trigger the processing) and at least one output
   b) That the flow name in should logically be connected to the processing that the process does and the name of the flow coming out should be reflective of the name(s) going in and the processed applied to it
   c) A process cannot exist if its only inputs and outputs are from and to data store. Stores are passive data they cannot trigger processing

### Question 1b

The major entities and appropriate relationships were identified by the majority of candidates and it was pleasing to see how many candidates identified the need for an 'occupancy' entity to store the data created and manipulated once a reservation becomes an actual visit/stay at the hotel. But, again a range of avoidable errors were noted, these include:

1. Including the physical buildings and staff involved in the scenario as entities. Entities called 'Clerk', 'Receptionist', 'Hotel', 'Chan Family' are not part of the **Logical** processing of this system about which we wish to store and manipulate data. Sometimes a useful test to apply to candidate entities is to ask two questions:
   a) Is it within the responsibility of this systems, ie. do we have a responsibility to do something to it (ie. apply a process)?
   b) Is there more than one instance of it?

   If the answer to all these questions is 'Yes' then it may be an entity in the system. A 'No' answer to any of the question should make you look more closely at the candidate entity and decide if it really is an entity or not.

2. More candidates than in previous years failed to apply the correct primary/foreign key posting rules. This is regrettable, because once learnt they can be applied with confidence and will produce a data model with the necessary referential integrity. The rules include:
   a) A 1:1 relationship that is obligatory at both ends requires that you consider merging the two entities into one.
   b) A 1:1 relationship that is obligatory at one end requires that you pst the primary key from the non-obligatory end to the obligatory end.
   c) A 1:M relationship that is obligatory at the M end requires the primary key from the 1 end to be posted to the M end.
   d) A 1:M relationship that is non-obligatory at the M end requires the primary key from both ends to be posted to a new entity that needs to be created to store the details of those instances that do participate in the relationship. This is similar to the decomposed M:M relationship explained below.
   e) A M:M relationship has to be decomposed into two 1:M relationship by creating a new entity between the two original entities to store the details of those instances that participate in the relationship. The primary keys from both of the original entities are posted to the newly created entity along with any other data that belongs more naturally to the relationship than at either end. For example a 'student' to 'module' relationship is likely to be a M:M, so we could introduce a new entity between the two called 'student/module' and include as data in the newly created entity the two primary keys and two other data items that do not belong at either end but which we do want to store, i.e., 'student module grade' and 'assessment date'.

**Question 1C**

Out of the candidates taking this paper at this sitting only one managed to get 10/10 marks and less than 10 others got between 1 and 9 marks. It is clear that candidates are aware of the context and top levels of DFD's but few know how these are then decomposed into levels of increasing detail that can then go forward to support later stages in the development process. Avison and Fitzgerald cover this topic very well in their chapter on 'Process Techniques'.

**Question 2.**

**2.** *a)* Identify **FIVE** fact finding techniques that may be used to gather systems requirements. **(5 marks)**

*b)* For the system outlined in Question 1, evaluate the suitability of **EACH** of the techniques identified in your answer to Question 2 *a)* for gathering the systems requirements. **(20 marks)**

**Question 2 Answer pointers**

**2(a)** Sampling; questionnaires; interviews; research; observation: 1 mark each.
Credit given for other reasonable answers, e.g. use of prototype, holding workshop.
(5 marks)

**2(b)** A brief description of each technique, then an evaluation of its suitability for the Chan family guest house:
Sampling: suitable; the hotel diary, booking form etc. will be valuable sources of information.
Questionnaires: not very suitable; hotel is small enough to be able to interview the staff. Could be used to gather guest views.
Interviews: suitable; key information will be obtained by talking to the hotel manager and staff.
Research: useful to obtain information about hotel booking systems.
Observation: useful to observe the dialogue with potential customers over the telephone, the check-in procedure etc.
(Up to 4 marks for each reasonable evaluation to a maximum of 20).

**Examiner's Comments**

This question, along with Question 5, was chosen by many candidates. It was designed to enable candidates to demonstrate their ability to *apply* their knowledge within a given context. Disappointingly, many answers *described* the five fact-finding techniques in great detail without *evaluating* them for their suitability for the Chan family guest house. No more than 2 out of the 4 marks available for each technique could be given for a detailed description that made no attempt to evaluate its suitability for the system. This highlights the importance of candidates' ability to accurately interpret examination questions.

**Question 3**

**3.** *a)* Identify the advantages of using prototyping during the **systems analysis** phase of a project. **(13 marks)**

*b)* Explain the issues that a project manager needs to address if prototyping is used during the systems analysis phase of a project. **(12 marks)**

**Question 3 Answer Pointers**

**3(a)** Requirements elicitation; early feedback from users to check analysts' understanding; help enlist users' support; aids communication with the user; help develop a system that meets requirements; aid user training later in the project lifecycle.
Up to 2 marks for these or other reasonable advantages to max of 10.
Plus up to 3 marks for definition of a prototype. (13 marks)

**3(b)** Need to plan carefully: either set a limited no. of iterations or use timeboxes to avoid project overruns.
Need to have available suitable software tools.
Need to ensure users understand that the prototype is an early version, that development of the full system may take considerably longer than the production of the prototype.
Up to 4 marks each for these or other reasonable suggestions to a max of 12. (12 marks)

**Examiner's Comments**

In answering (a) candidates tended to list the advantages of using a prototyping approach within the entire systems development lifecycle; referring for example to code generation or reverse engineering. As a question on the Systems Analysis paper, the focus was clearly on the advantages of using prototyping during the **systems analysis** phase. Marks were lost by candidates who did not directly answer the question.

Part (b) was seeking those issues that the project manager must consider *if prototyping is used* – some candidates listed the project manager's responsibilities in general, instead of focussing on the context of the question.

**Question 4.**

**4.** *a)* Explain how functional requirements are modelled and documented in the systems analysis phase of a project using an Object-Oriented approach to systems development. Use **ONE** of the processes identified in Question 1*a)* to illustrate your answer with a suitable example. **(13 marks)**

*b)* Explain the role of the Class Diagram in the systems analysis phase of an Object Oriented project. (You are **NOT** required to provide a diagram.) **(12 marks)**

**Answer Pointers**

**4(a)** Documented using:
use case diagram – a high level summary of all functions, together with the external entities (Actors) with which the system interacts, and
use case descriptions – textual description of the user function represented by the use case.
Up to 4 marks each for suitable explanation = 8

Up to 3 marks for an appropriate diagram and 2 for a use case description of a process identified in the answer to question 1 = 5
Credit will also be given for appropriate explanation of activity diagrams. (13 marks)

**4(b)** Shows all the classes required to provide the functionality specified in the Use Cases; shows the data (attributes) and behaviour (operations) that are the responsibility of each class; shows the associations between classes (i.e. the way in which they will interact to provide the functionality).
Up to 4 marks each for these or other relevant explanations. (12 marks)

**Examiner's Comments**
This was the least popular question, perhaps indicating candidates' lack of experience or confidence about object-oriented analysis. Those who did answer the question mentioned the use case diagram as a means of modelling requirements, but omitted their documentation as use case descriptions. The latter are an essential component, without which the diagram is incomplete.
In part (b), candidates recognized the role of the Class Diagram in identifying the analysis classes and their attributes and behaviour, but frequently failed to mention that it also indicates via associations the way in which classes will interact to provide the functionality documented in the use cases.

## Question 5

**5.** Briefly explain any **FIVE** of the following terms in the context of systems analysis:

*a)* Feasibility Study
*b)* Walkthrough
*c)* Timebox
*d)* CASE (Computer Aided Software Engineering) tool
*e)* Non-functional requirement
*f)* Stakeholder
*g)* Rich Picture. **(5 x 5 marks)**

## Question 5 Answer Pointers

### a) Feasibility Study
Study conducted to determine economic, technical and operational feasibility of the proposed system; carried out before the systems analysis phase; brief description of contents.

### b) Walkthrough
A means of quality control; peer review of work to date; provides constructive criticism to improve quality of systems analysis effort; requires preparation and follow-up to be effective.

### c) Timebox
A specified amount of time within which a task/phase is to be completed; the deadline is immovable; a means of ensuring projects deliver on time; cut down on objectives/work completed rather than overrun.

### d) Computer Aided Software Engineering (CASE) tool
Software support; enables the production of models required for systems analysis; models are kept in a central repository available to all project team members; can be more easily kept up to date; consistency, completeness checks can be made; can be used to generate systems documentation.

### e) Non-functional requirement
A systems requirement that is not part of the functional specification; examples include preformance, ease of use, cost, maintainability, portability.

### f) Stakeholder
All those likely to be affected by the system, either directly or indirectly; include end users, team members, resource managers, recipients of system output; local communities/public at large.

**g) Rich Picture.**

A technique used within Soft Systems Methodology; shows the 'richest possible picture' of a problem situation; includes overview of the situation, main stakeholders, processes, information flows; highlights concerns and conflicts; identifies external scrutiny.

1 mark for each relevant point made/each relevant example to a max of 5.

(5 marks each x 5 = 25 marks)

**Examiner's Comments**

This was the most popular of the optional questions and overall there were some very full, accurate answers. Candidates should recognize that when asked to answer any FIVE this means that only five answers will be marked; sometimes time (and marks) were wasted by candidates answering more than this number of parts of the question. Candidates would also do well to put approximately equal effort into their answers to each of the five parts, rather than write a good deal about one or two, and run out of time to write more than a few lines for the others.