In total, 79 students sit in the exam.

Question 1: taken by 77 students, and the average mark is 65%;
Question 2: taken by 61 students, and the average mark is 54%;
Question 3: taken by 71 students, and the average mark is 64%;
Question 4: taken by 12 students, and the average mark is 53%;
Question 5: taken by 16 students, and the average mark is 51%.

Further observations:

(1) The most common mistake is in question 2: digital certificates or public keys are not confidential information – they are public information, therefore their transmission does not require confidentiality protection, rather it requires integrity/authenticity protection against unauthorised modifications and impersonation attacks.

(2) In total, 3 students fail to pass the 40% mark; 27 (i.e. 34%) students get results in the range of 60-69%; 20 (i.e. 25%) students get marks >= 70%.

(3) There are some tough bits in the paper, but students have generally done well. I am very pleased to see this.
Much more detailed feedback is available for this exam via the Assessment 21 feedback tools on-line.

Q1.
Slightly low average for this question.

1.1
Generally good answers. Some students restated the obvious wired not wireless which was not worth a point unless this statement was considerably expanded. High average for this section.

1.2
Some good "parked" answers. Several confused this with Bluetooth scatter nets which partly answers and was given 1 mark. Lots of students did not answer this section correctly.

1.3
Too many answers assuming wired plain old telephone system (POTS) for 1G systems rather than “mobile”. Too many students missed the "reasons" keyword in the question and therefore lost or failed to pick up marks. Reasonable average.

1.4
Adapted marking scheme to 2 marks for anything vaguely relevant. 2 points for any example. To maximum of 6.

Demonstrated that most students still have little idea what multi-carrier modulation is and no idea what is is used for or why. However, some good attempts. This separated the best students effectively, weaker students mostly did not answer this section. Very low average mark.

Q2.
Above target average for this question.

2.1
Some think propagation through walls is better at 5Ghz than 2.4GHz but evidence states there is not much difference. Did I say this? Hopefully not. Reasonable average.

2.2
Some good calculations and some silly mistakes mainly with magnitudes. Not enough students showing working even with strong reminder in the question text. Reasonable average.

2.3
Mainly good answers with a high average for this section.

2.4
1/2 marks per point allocated. Max 2 (4 * 1/2) per type of orbit. Minus 1 for missing orbit type(s). Generally good answers but some students left out one or more orbit types that were taught. Highish average.

Q3.
Slightly low average for this question.

3.1
Mainly sensible answers to this. Reasonable average.

3.2
Lots of information missing on multi-media data rates using GPRS and why GPRS is not good enough. Slightly low average.

3.3
Reasonable answers with a reasonable average.
3.4
Mainly good answers but too many students knew one half or the other of this question. Low average as a result.
As in previous years, I have to comment on the scruffy presentation of answers by some of the candidates. Not only was the writing often illegible (just a scrawl), but the answers to individual questions were quite often interleaved with no indication of that being done. The quality of English in many of the requiring textual explanation was also rather poor.

Q1:
This question was a modelling question, aimed to discover the student's ability to understand and modify a simple concurrent system application written in FSP from a natural language description of requirements.

4 students answered the question. Parts a to c were answered largely correctly. Part d, which carried 9 marks, required the modelling of a countdown timer and its incorporation in the previous model, and modelling what happened when the timer ran out before the gates had closed. 3 out of the 4 students answered this part: all arrived at a reasonable description of the countdown timer, but all lost marks through not handling the situation following "timeout" appropriately.

Average mark 14.

Q2:
This question was aimed to show an understanding of traces and the composition of processes and properties in FSP.

26 out of 45 students answered the question. Parts a, b, c and d were largely answered correctly -- but some students misunderstood the meaning of a process parameterised by a value, e.g. VM(15), incorrectly interpreting it as an indexed process VM[15]. Many answers to part e were then incorrectly constructed, because of the above mistake. Part e also asked for an explanation why the given composition yielded the possible servings --- few answers appeared for that. Part f was also poorly answered by most: whilst it might be possible to sit down with the LTSA tool and work through all possibilities (certainly the case for the small example), in general this is not a sensible direction - the question was seeking an understanding of a safety property, that would automatically give the answer.

Average mark 12.

Q3:
This question was aimed to show an understanding of bisimilarity between FSP processes.

40 out of 45 students answered the question. For the most part, the question was answered well and completely. A few answers showed very poor understanding of the bisimulation concept --- indicating little or no revision of the particular topic. A number of answers showed a conceptual misunderstanding that a colouring algorithm could be applied separately to the labelled transition systems of a given pair of processes, rather than being applied to the two distinct LTSs considered as just one system (crucial in order to get the same colouring of equivalent states). Some answers were too vague with explanations of the difference between bisimilar processes and trace equivalent. Some answers, whilst they exhibited examples of processes that were indeed trace equivalent but not strongly bisimilar, did not give explanation of why the processes behaved differently.

Average mark 14.

Q4:

30112 Concurrency Howard Barringer

06 October 2010
This question was aimed to show an understanding of concurrency concepts in Java.

30 out of 45 students answered the question. Overall, this question was answered poorly. Part (a), in particular, was badly undertaken: marks were lost through misunderstandings, vagueness and confusion in answers. For part (b), most answers failed to give the correct reason for the inclusion of the alphabet extension, namely to ensure that write[0] was in the alphabet of INC and that write[N] was in the alphabet of DEC, as without VAR would be free to set to zero during an increment, etc. The description and explanation of the FSP was, in general, poorly given. Many of the answers to part (c) gave no explanation of the presented Java outline code; several answers showed a misunderstanding of the intrinsic locking via the synchronised methods; several answers failed to implement the model as presented, allowing unbounded counters or, even, increment methods that may not increment.

Average mark 10.7

Q5:

This question was aimed to show an understanding of safety properties in FSP and LTSA, and use of semaphores

35 out of 45 students answered the question. For part (a), many answers seemed to ignore the fact that the question asked to explain the difference between a blocked process, deadlocked and livelocked system in terms of FSP, and were often couched, for example, in terms related to the notion of a blocked thread in Java. Part (b) was mostly answered correctly, as was parts (c) and (d). The first question in part (e) required a simple relabelling: most students realised this. The second question of part (e), which required a complete rewrite of the dining property to allow for all interleavings of the sitdown and getup actions for up to two possible diners, caused difficulty for a large number; there were some odd wrong attempts, for example the inclusion of two copies of the property, that showed poor understanding of synchronisation and process/property composition in FSP.

This was the final question answered by many students and it was clear that several ran out of time.

Average mark 11.6

Comments:

30142 Compilers Rizos Sakellariou

Q1: 13/21 students answered this question. No particular problems. In (c), several students failed to describe the output of lexical analysis (tokens), or parsing (Abstract Syntax Tree). In (d), only a few students noted the impact of the complexity of the register set in modern processors.

Q2. 19/21 students answered this question with overall good marks. Many students failed to use Thomson's contraction in b.i even though it was explicitly requested. A common problem in the conversion in b.ii was that the use of a not very compact NFA caused problems. Some students showed a DFA which clearly couldn't accept the input expression.

Q3. 15/21 students answered this question. The average was somewhat low. A common problem in (a) was that some students didn't show the parse tree. In (b), the transformation required to get rid of both the recursion and to guarantee the LL(1) property. Few students did both.

Q4. Only 4 students answered this question.

Q5. 10/21 students answered this question. A common problem in a.i was that student didn't take the most frequently used values to allocate to physical registers. In (b), only few of the answers examined different weighting functions to cope with the heterogeneity of the functional units, and very few considered the issue of how to choose (using an algorithm not intuition) a functional unit for an instruction.
Q1. String matching algorithms

The Boyer-Moore string searching algorithm was described with varying degrees of accuracy. Those who understood the BM method lost marks through incomplete descriptions of the two preprocessing phases (including how the information about the possible text character at a fail point is stored), and in the failing to say clearly that the maximum of the two shifts computed at a fail point is used. Some students clearly had only a hazy idea about the algorithm - and got correspondingly low marks.

The suggested algorithm based upon preprocessing the pattern to determine matching positions for the pattern against a text character is sometimes called "bucket matching". Most understood the general method and could describe the preprocessing phase, but lost marks through imprecise descriptions (we need naive character-by-character matching and then at each fail, attempt to re-align the pattern to the current text character). Some didn't read the description carefully enough and thought that this was another algorithm (eg the KMP algorithm) - no marks were awarded for this.

Q2. Graph Algorithms

This question was attempted by many of the students and overall the marks were good. The description of DFS techniques for graphs, including the (pseudo)code was well answered by most. The modification to include ancestry information: one method is to use a stack - pushing when encountering a new node, popping when backtracking (and an additional hash-table for quick lookup in the stack). Answers that included this basic idea (or others that work) gathered marks, others who thought that dfsnum alone was enough lost marks.

The final parts on strong components and DFS: Some students didn't understand that strong components, like connected components, partition the set of nodes. Approx half realised that for a graph consisting of a single strong component, the DFS forest consists of a single tree. As for the question about multiple runs with random DFS starting points, most students gained only few marks. Half marks were awarded for noticing that the number of trees in such forests is bounded above by the number of strong components. The graph structure of edges between strong components is acyclic. Properties of this graph can be inferred from the numbers arising in a sequence of runs.

Q3:Q4. Unfortunately, yet again, I have to comment upon the scruffy presentation of many of the answer scripts. It is appalling that we have students at this final year level who appear incapable of presenting answers in anything but an almost illegible, incoherent, scrawl. It seems the school needs to take a much tougher stance earlier on in order to help students present their handwritten work and, hence, themselves, in a clean, clear and professional way --- the school shouldn't have to do this, however, it clearly doesn't happen properly at secondary education level.

Q3:

This question was aimed to show some understanding of basic model checking principles and its relation to graph algorithms.

52 out of 71 students answered the question. Apart from a few very good answers, this question was surprisingly poorly answered. For part (a), many answers were too vague, didn't address the key concepts and perhaps mentioned just one feature to justify the success of model checking. Most answers showed knowledge/memory of a Kripke structure but only a few gave good explanation of how the behaviour of a concurrent shared memory program can be encoded as a Kripke structure. The example construction of a Kripke structure for the sample program was answered well by a reasonable number; it is easy, however, to make mistakes and whilst allowance can be made in some cases, it was clear that some answers did not understand the construction process (probably indicating that they hadn't attempted sufficient of the ample exercises provided in the lecture notes and past examination papers).
Most answers gave reasonable outline of an algorithm to compute the forwards reachable set of states from a given state; there were, however, several answers which showed little understanding of how such an algorithm can best be applied for the particular question.

Average mark 11.

Q4:

This question was aimed to show some understanding of infinite word automata and related graph algorithms.

35 out of 71 students answered the question. There were a number of very good answers, which showed good understanding of Buchi automaton acceptance conditions and the product operation on such automata. A few answers showed very little understanding. The first part of the question on strongly connected components was well answered by the majority, although there were some who did not read the question carefully, e.g. giving examples of SCCs and terminal SCCs from the course material rather than addressing what was asked for by the question (a two state graph showing both). Nearly all students managed to give automata that accepted the given languages, though some had many more states than necessary. Again, it is likely that those who managed this part well had undertaken the course exercises rather than just reading lecture notes. Practice is necessary to ensure an understanding of the process.

Average mark 12.1

Q5: NP completeness

Few students attempted this: marks tended to be either high or low!

The first half of the question was bookwork/lectured material and there was no excuse for some of the poor answers, except lack of preparation for the exam and lack of engagement with the course. Those who put in the effort got correspondingly good marks.

The second part was well answered by those who had followed the NP-completeness proof in the lecture, and those in the exercises. It required the NP-completeness of the decision problem for "independent sets of nodes". Many observed that this is the clique problem for the "graph complement". This gained some marks - and noting that complementation is polytime. Using this to code a reduction proof from propositional satisfiability is fairly straightforward - again marks were lost through incompleteness of answers and imprecision.

This exam was sat by a very small number of students (6) and thus it is difficult to draw any meaningful conclusions. Overall the exam performance was disappointing, particularly as those students who did attend the lectures were keen and interacted well, asking intelligent questions.

One reason for the poor marks was that question 1, which although not compulsory was answered by all students and in general answered very badly. This was a surprise because I thought this was the easiest question and covered material - the use of pass transistors - that occurred in many parts of the lecture course. It may well have been this very fact that caused most of the student to mis-interpret parts of the question and give erroneous answers. For this reason, should the examiners feel int necessary, I would not object to the marks for this course being upcaled.

 Comments:

This exam was sat by a very small number of students (6) and thus it is difficult to draw any meaningful conclusions. Overall the exam performance was disappointing, particularly as those students who did attend the lectures were keen and interacted well, asking intelligent questions.

One reason for the poor marks was that question 1, which although not compulsory was answered by all students and in general answered very badly. This was a surprise because I thought this was the easiest question and covered material - the use of pass transistors - that occurred in many parts of the lecture course. It may well have been this very fact that caused most of the student to mis-interpret parts of the question and give erroneous answers. For this reason, should the examiners feel int necessary, I would not object to the marks for this course being upcaled.
Comments:
q1. answered very well
q2. answered very well
q3. answered very well
q4. answered very well
q5. not answered
Question 1

a) Most people achieved full marks for this question. Where marks were lost, it was for poor or incorrect justification (e.g. stating that the second change was adaptive because the organisation had to adapt to a new operating system). In a small number of cases, candidates appeared to be unfamiliar with the meanings of the different types of change and seemed to be guessing as to their meaning. These answers earned few marks, since the guesses were not close to the true meanings.

b) There were a small number of perfect answers to this question, and a great many candidates achieved high marks for DDGs with just a few missing or incorrect elements. A surprising number of people added a separate node for the code on line 25, even though it is part of the statement that begins on line 24 and so does not require its own node. This mistake was made in several cases where the graphs were otherwise perfect, as well as in much poorer graphs and in graphs where other multi-line statements were handled correctly. To repeat, the nodes in a DDG correspond to statements in the code, not line numbers. We use the line number of the first line on which a statement appears to label its node, merely as a convenience for human writers/readers of such graphs.

A similar error was made by a number of people who included nodes for lines 8, 16 and 17.

Other common errors include:

+ failing to include the recursive dependencies on nodes 19 and 20,
+ forgetting that the dependencies are often different for first time the loop is executed, so that dependencies to nodes 7 and 9 were omitted, and
+ giving a dependency from node 18 to node 10, which does not exist.

c) Apart from a small number of people who seemed not to know what a control dependency was, and provided odd unidentifiable graphs, most people managed to work out the main dependencies here. Some people lost marks by not giving the transitive dependencies (as asked for in the question), while others gave dependencies between statements 18, 19, 20 and line 17 (the "else" of the "if" statement that begins on line 12). A significant number of people omitted the recursive dependency from statement 11 to itself - a common error when CDGs are calculated by eye, rather than by following the full procedure.

d) There were few good answers to this question. Most candidates waffled about some of the software quality issues raised during lectures and the revisions sessions, without going into much detail or depth. These answers typically earned around half marks for this question part. Fewer marks were earned by the (sizeable) number of candidates who claimed that a disconnected sub-graph in a DDG definitely indicated a problem with the code (followed by some hand-waving about low cohesion). This is incorrect. Disconnected sub-graphs may indicate a problem, but the code they represent may be just fine, since control flow links may connect the sub-graphs into a meaningful whole. It is certainly not the case, as one candidate suggested, that we can remove the for-loop from the method in appendix A just because there are no data dependencies on l in the rest of the code.

Question 2

a) Although there were a number of good answers to this question, many candidates talked in general about top-down (or other) reading strategies without relating their answers to the specific scenario mentioned in the question. Such answers earned typically only about half the marks available for the question as a whole.

B) Again, some excellent answers were given to this question, but there were also many answers revealing misunderstandings of the general principles of code reading. A very common mistake was to point to individual code statements rather than complete idioms. For example, referring to the call to the strcasecmp2 function as a "comparison idiom" or the "i++" as an "increment idiom". The whole point and value of the concept of a code idiom is that it is something larger and more complex than a single statement or method call. It expresses some commonly useful package of
behaviour that requires several statements to implement, but which must be recognised quickly during code reading. "i++" by itself is not an idiom, but it may be a part of a larger idiom (such as process every item in a collection). Some of the strangest attempts at idioms given were the "if loop (???) and the "comment idiom" (which referred to a single comment line in the code).

A less serious (and less common) error was to describe the correct idiom, but in a less than general form. Many candidates mentioned specific data structures (e.g., naming the idiom as "search an array for ..." or "search a linked list for ..." instead of "search a collection for ..."). The power of idioms is in their generality, so it is important that they are not tied down to specific languages or data structures when not necessary.

A number of candidates thought the code included examples of the "process every item in a collection" idiom. Since the two loops both exit from the entire function once an element with the correct property is found, it follows that in many cases only a portion of the collection will be visited by the loop. These answers lost two marks for naming the incorrect idiom, but were otherwise marked as if "process every item in a collection" was present in the code.

Several candidates thought they had identified an example of the swap idiom, in the call to strtol/3. Presumably, the idea here is that the function and assignment swaps the value of svc with id? Certainly, id is assigned a form of the value of svc, but the svc variable remains unchanged by the operation. There is also no obvious temporary variable, as is needed by the classic swap idiom. So, this idiom is not present in the code.

Finally, a number of candidates were tripped up by the fact that strcasecmp returns "FALSE" when a match is found, not "TRUE" as might more intuitively be expected. (This was explained in the text which introduced code fragment A.) Therefore, the "if" statements on lines 18 and 24 test for the existence of a match, despite the negation operator (!) in their conditions.

C) There were two main causes of lost marks for this question. One of these was a failure to read the text of the question carefully, so that only a part of the answer requested was given. Candidates who fell into this category provided a hypothesis as to the behaviour of the function, but failed to note any of the questions or conjectures that led them to formulate the hypothesis.

The other main cause of lost marks relates to the description of the hypothesis itself. One of the most important skills in code reading is being able to relate individual lines of code and data structures to the human-oriented tasks and actors that the software supports. This part of the question was aimed at testing candidate's ability to do this. But some candidates merely gave a low-level description of what the individual parts of the code were doing, rather than giving a high-level description of the behaviour of the function as a whole as it relates to Android as a whole. For example, the answers which scored high marks for this question talked in terms of searching for "services" (e.g., "nearby services" or "registered services" - the code tells us only that they are "BNEP" services, but any plausible guess such as these earned marks) rather than "searching the __svc array".

A small number of candidates misread the text which introduced code fragment B, thinking that the values in the __svc array were the reserved names, rather than that the name of the array itself (svc) was reserved. (I.e., it is a reserved variable name.) Where this mistake occurred, a mark was removed for the mistake but the rest of the answer was marked as if the candidate had correctly interpreted the term "reserved" in the text.

D) Answers to this question also suffered from the mistaken understanding of the term "idiom" described in connection with part b) above. Some candidates failed to describe a general-purpose idiom, but instead merely gave a description of the details of what the code was doing. Other candidates pointed to individual code statements or operators as idioms. For example, pointing to the use of the Log.v method and calling it (all by itself) a "logging idiom", or describing the semantics of the "if" statement in Java.

A small number of candidates did not read the question carefully, and so gave their answer in terms of code fragment B, rather than code fragment A as requested. Where this happened,
two marks were lost for the mistake, but the answer was otherwise marked as if code fragment B was the intended source for the new idiom.

Most of the candidates who provided a true idiom pointed out the presence of a version of the "process every element in a collection" idiom, which included provision for special/different behaviour in the case of the first item in the collection. Only a handful of candidates pointed out the apparent strangeness of the implementation of the idiom. Usually, if special behaviour is needed for the first item of a collection, it is encoded before the loop body, and the loop begins with the second item.

The encoding given in fragment B, however, is one of those used when the collection may be empty (and may therefore have no first item). Since it would seem strange to run a stress test with no executions of the camera start-up routine, the implementers must have been coding defensively here, just in case some future tester (or tester software program) decided to set TOTAL_NUMBER_OF_STARTUP to zero.

Question 3

a) Almost everyone earned full marks for this question, making very sensible calls on the decomposibility of the two systems. Only a handful of candidates remarked on the complications caused by the fact that the Delivery Planner seems to use presentation logic from the Sales and Delivery Handler but bypasses the data access component. This suggests it contains both application logic and data access code.

Where marks were lost, it was for not including all the information asked for in the question.

B) There were many excellent analyses provided in answer to this question, with strong cases being made for both forward and reverse engineering. However, there were many candidates who lost marks because they talked only in general terms about the relative strengths and weaknesses of the various strategies, rather than relating them specifically to the scenario. For example, if an answer claimed that general migration is not suitable because it requires complex gateways then that would have earned no marks. But, an answer which remarked on the complexity of writing gateways to connect to the four databases of the scenario would have earned marks.

The question of what new data/functionality is needed by this example is a tricky one, and few candidates commented on this issue. As many candidates claimed that new data was needed as claimed that it was not. Similarly, and importantly for this question, the issue of whether the required functionality could be supported by a simple integration of the existing databases (through a gateway) was not discussed, despite the fact that it is crucial to whether a reverse migration strategy is possible or not. The central questions to be asked are:

+ Can delivery plans for Mud Unlimited products be added to the InterFlower database, through a suitable gateway?
+ Can Mud Unlimited products be made available for sale through the InterFlower Sales and Delivery Handler, if a suitable gateway makes it appear that they are present in the InterFlower database?
+ What is the complexity (costs/risks) of such a gateway?

+ Can the functionality to be provided by the new GUIs be provided by a combination of calls to the legacy components, given suitable gateways to orchestrate that combination?
+ What is the complexity (costs/risks) of such gateways?

If the answers to these questions suggests that overly complex gateways are required, then the reverse migration strategy is not appropriate.

For forward migration, the analysis is much simpler. The questions are:

+ Can the new merged business operate sensibly using the existing legacy components, if suitable gateways can be created to make it appear as if each legacy component is running over a database that contains the contents of both business’ database?
+ What is the complexity (cost/risk) of those gateways?

As every year, there was a number of candidates who had misunderstood the term "general migration", and therefore proposed it as a "best of both worlds" strategy, without any
discussion of the associated complexities and challenges.

A small number of candidates mixed up the terms forward and reverse migration. After a 2 mark penalty, these answers were marked as if the correct terms had been used.

C) There were some excellent answers to this question, and a number of candidates managed to achieve a first class mark for the question as a whole, despite the fact that their answer to this part was only half finished.

However, there were also many answers which contained very odd migration patterns. Some of the commonest mistakes are listed below.

+ In some of the migration steps, users were expected to access application layer components directly, without any intervening client software. It was not clear from the answers how the candidates expected this access to take place. Perhaps these steps were meant as intermediate versions of the architecture that would never be in real use? If so, the candidates should have saved time and not drawn them. Only the versions of the architecture that will be deployed (or enter significant user testing activities) should be drawn.

+ Some migration steps required users to access the system directly through a gateway - even in some cases to access a GUI through a gateway. Similarly, some migrations placed gateways below data access components, presumably to allow them to connect to a new/old database. Neither of these is correct. Gateways occur only between layers. It is not correct to have a gateway above the presentation layer or below the data access layer.

+ Failing to state what type of gateway was placed into the architecture. Where steps were described using text only (no diagram), it was commonly the case that the number and placement of gateways intended was not described by the text. It was very difficult to award marks for gateway usage for these answers.

+ Migrating target GUIs into the architecture, without considering how (whether) the legacy components will support the functionality they provide the front-end for. Some migration plans even involved beginning by migrating all the target GUIs in one step, before any of the support functionality is definition in place.

+ Giving a sequence of steps that implement one migration strategy when a different strategy had been chosen in part b). Most commonly, candidates who said they preferred a general migration strategy often ended up giving migration steps that followed forward migration.

Even in the best answers, it was the case that the capabilities and connections of the legacy components were not really taken into account. For example, several candidates talked about the need to get the sales functionality "up and running" as fast as possible, so that the company could make money. But the whole idea of the incremental migration is that the sales parts of the system should always be up and running, initially supported by the current legacy systems and later by the target components. Therefore, this in itself was not a good reason to prioritise the sales components for migration (although there are other good reasons why you might want to do this).

A further issue relates to the number and placement of gateways between the application layer and the data layer. Many candidates chose to implement a forward migration, which required that forward gateways be installed between the legacy components and the new data access components. Some candidates chose to create just one gateway, that took input from all the legacy components and communicated with both new data access components. This is possible, but since the commands issued by the legacy components are in quite different formats/protocols, it means that a complicated gateway capable of understanding both protocols is required.

Another configuration that was provided by a number of candidates was to create one gateway for each of the new data sources and to connect the legacy components to both these gateways. But this cannot work, as it requires the legacy components to have the ability to choose which gateways they send which requests to. We don't want to have to change the legacy systems any more than we have to (to create the maximum number of
decomposed units) and adding code that is specific to gateways, and therefore temporary, is a particular waste of time and effort.

The proper configuration is to create one gateway per legacy component that needs to talk to the data components. These gateways can then have the task of deciding which of the new sources they need to communicate with. (If two or more legacy components issue/receive requests using the same protocol, then it may be possible to reuse a single gateway for all of them. This is not the case in this scenario, however.)

Finally, a surprising number of candidates made very strong statements about the low level of risk involved in implementing and deploying various of the components, despite the complete lack of evidence for such claims. Candidates talked about the risk being "minimal", "negligible" and "very small", even for known-to-be-risky tasks, such as merging the two databases. This unrealistic assessment of risk was a cause of some unconvincing migration choices, and therefore of some lost marks.

Question 4

This question is normally very unpopular, but this year a sizeable number of people attempted it (though it was still the least popular of the questions).

A) There were several excellent answers to this question. Where marks were lost, it was largely for proposing questions that were too similar to each other. For example, some candidates gave a long list of very similar questions covering the different data types. "How often are diagnoses made? How often are drugs prescribed? How often are hospital procedures performed? Etc." These long lists scored just 1 mark, because all the questions were about eliciting the same kind of information. To determine the frequency of a feed properly, it is necessary to ask about a range of different aspects of the system (system capabilities and freshness requirements, for example, as well as data dynamics).

B) It was clear from some of the answers that candidates were guessing the meaning of the terms "push" and "pull", based on their meanings in other parts of computer science, rather than having a clear understanding of their meanings in the context of ETL strategies. Other causes of lost marks for this question include failing to relate the answer to the specific context described in the question (i.e., giving generalities about the different strategies without making their relevance to the NPR DB context clear), and simply saying too little to warrant full marks.

C) Most candidates gave good answers to this part of the question, and were able to make clear cases for their preferred migration ordering of the data types mentioned. Where marks were lost, it was because insufficient justification was given or because not all of the data types were mentioned.

D) The most common error made in answering this question was to refer to schema-level inconsistencies (in particular, having different data types in the source databases for a given piece of data) rather than data inconsistencies. Other candidates gave one of the two pieces of information requested (the type of inconsistency and an example) but not both, thus losing marks.
Overall, this question was answered reasonably well.

A key aspect of 20 questions is that expert asks the knowledge engineer questions, thus revealing the aspects that the domain expert considers to be important. This was missed by some candidates.

Answered well in general. There were some common modelling errors, such as including "University" as a top concept. Identification of modifiers was not always well done.

Essentially recall.

Some answers considered that the problem was related to the possible existence of a "one-man band" which would then not be a group of people (although there is no reason to disallow singleton groups). The answer expected was based on the presence of an identity criterion. A Group of people has an identity based on the people in the group. This identity criterion will not hold of Bands though, as the same group of people could form a different band. Thus a Band is not a subclass of GroupOfPeople, but should be considered to be made up from a group of people. This follows the example of Social Clubs as discussed in the OntoClean papers.

A spread of answers here. Some candidates identified the distinctions between navigational structures and intensional definitions. The possibility (or not) of using inference is also a consideration. Answers receiving top marks were those that not only discussed the differences, but placed those differences in the context of the task in hand.

Bookwork

This question was answered less well, in particular w.r.t the extra-logical services. Examples of extra-logical services that were discussed during the course include the use of OntoClean (not concerned directly with the underlying formal semantics of the representation), and annotation, for example with lexical labels.

In general, we expected a bit of detail sufficient to "contrast" the foci. E.g., commonsense knowledge tends to be broad, shallow, and incomplete, whereas scientific knowledge tends to be narrower, deep, and aims for completeness. For example, it's part of our commonsense that a robin is a bird, but "minimally" a scientific representation distinguishes between an American and European robin and distinguishes them on very specific grounds (they are members of the same order but not the same family). Most people got this correct.

Only a few people got this completely right. The key distinctions are that a data structure may contain representation irrelevant aspects (e.g., bookkeeping variables) and has no inherent representational reading (*any* part of it might be representational; there is no built in even *formal* semantics).

The roles of medium of efficient computation and medium of human expression were discussed extensively both in the specific lecture and then again throughout the course and were called out by role number many times. Under this nomenclature they were clearly part of the technical lingo of the course. Most people had no trouble with this.
Q2.2a Most people got this correct. In general, the greater the expressivity of the logic, the more computationally difficult it is to reason with.

Q2.2b We were fairly generous with the specifics of notation, pretty much accepting anything that was discernably symbolic. Several people forgot to categorize the axioms as TBox or ABox or got that distinction wrong.

Q2.3a The answers were generally quite good here.

Q2.4a There was a lot of variance on this question. The most obvious benefits discussed in class is that description based development allows for e.g., automation verification of the representation (as well as reducing e.g., the number of "links" that must be manually maintained and globally inspected) and that post-coordination requires such development like features at runtime.

Q2.4b Only differences (definitions/descriptions, lack of prescription, open world assumption, model theoretic semantics, lack of methods) or similarities (inheritence, slot like structures) were accepted, though some from each side were preferred.

=== Q1.4d ===

Answers receiving top marks here were those that explicitly considered the web aspects of the representation, e.g. the use of URIs for concept identification, and building on existing infrastructure such as XML. The open nature of OWL is also important here, as is the ability to modularise ontologies.
Encouragingly this was one of the best sets of responses for many years, with many students giving plausible answers to most questions.

A few common misinterpretations occurred:

Question 1:
c) The intention of the question was to focus on 3D medical images (as described in the original lecture on registration). However, some students (reasonably in retrospect) interpreted it to refer to photographs of faces. Full marks were given where this was clear and a sensible answer was given.

Question 3:
a) The question was looking for an answer including descriptions of general interest point detectors (e.g., Corner detectors, blob finders etc.) which might fire on wheels. Almost all students interpreted the question as “how do you look for wheels” and described the circular Hough transform. Full marks were given for a sufficiently detailed description of the Hough transform for circles.

Question 4.
(a) Edges detection. Generally well answered. Most students described a Sobel or Canny edge detector sufficiently well. The main problem was that the question asked for location of the edges. This requires the gradient to be thresholded (or something) to generate a binary version. A number of people didn’t include that step.
(b) Generally fine descriptions of the Hough Transform. Since it is a straightforward question I wanted the $\rho-\theta$ rather than the $m-c$ representation.
(c) Most people got the issue about scanning normal to the edge and interpolating gradient values at non-integer grid points. The point that many missed is that the edge location requires a model of the edge profile across the edge to find the maximum.
(d) Despite the fact that the question is all about finding edge positions, a number of people lost marks by regurgitating lecture material about finding correspondences using interest points and cross correlation, which was irrelevant.

Question 5
This was generally well answered. Most people understood the image processing operations pretty well.
Part (f) was looking for some insight into the problem. People reproduced descriptions of active contour models and got marks according to how much detail was provided. However, full marks were obtained by noting that the long fibrous features would need an open-ended active contour, with ends fixed, and that the end points could be defined interactively to initialize the search. A few people spotted that the image energy term should try to respond to ridge-like, rather than edge-like features.
Q1. Covered the topic of Distributed Databases and Oracle. The statistics show that this question was solved by 50% of the students who sit the exam, and the average mark, considering only these students, was of 62%, which is reasonably high. Obviously, students benefited from examples on slides and a tutorial given during the lecturing period, which contained similar questions. Most students failed to solve the most challenging part of the question, which requested students to write a complex SQL statement, indicating that their Databases background is weak.

Q2. Most of the students attempted this question (85%, avg. mark 63%). In part (a), most marks were lost by not discussing CUBE/ROLLUP extensions to support drill-down. In part (b), marks were lost due to occasional errors in how to calculate confidence (dividing by support of the right-hand side). Part © was challenging for many, in particular in defining what the role of training is. Most of the students explained the ID3 algorithm well. Overall, there were not so many weak answers to this question, but there were a couple of very high ones.

Q3. Only half of students attempted this question (avg. mark 63%). Part (a) was easy (book-work), and most students got it right, but still there were cases where this was left unanswered or very general discussion on data warehouses was presented. Part (b) required a bit of computation, and some students noted and used the fact that there was no need to calculate sqrt for any distance. There were occasional errors in doing only one iteration of k-means or in assigning points to clusters. Surprisingly, square error estimate was not done correctly by many students, and many have not attempted it at all. Part © asked for discussions on a specific example of a business, and surprisingly there were many cases where the students just discussed general properties of data mining and OLAP, without giving context to their discussion. Most of marks were lost in such cases. Overall, there were a couple of very high marks earned on this question.

Q4. This question was attempted by just over half of students, and the results were good (avg. mark 72%). Most of marks in part (a) were lost on not explaining the two-step approach in sampling and partitioning. Part (b) required AVG function in SQL; marks were lost by using only ROLLUP. Part © was a success, with many students getting it right. Some marks were lost by just ‘guessing’ precision and recall. A number of high marks were earned on this question.

Q5. Again, half of students attempted this question, but this time with an average mark of 59% (due to a couple of very low marks). Part (a) was generally answered by most students, although some struggled to find similarities between data mining and OLAP. Part (b): in some cases, distances between objects and not between clusters were discussed; those did not bring any marks. There were not many discussions on effects of intra-cluster distances, so not many marks were earned there either. In part ©, most marks were lost in not specifying the schema properly (e.g. by not linking dimension tables to the fact table). In some cases, it was not clear what the measures in the fact table are, as some solutions included various dimensions directly in the fact table (e.g. phone type).
Examiner comments on COMP37342 Advanced Topics in Information Retrieval examination, 2010

General: The following statistics refer to unconfirmed marks. Performance was in general good, with the average mark being 57.50% (18 candidates sat the examination). The highest mark awarded was ~87%. The lowest mark awarded was 30%. Question 1 averaged ~63% (14 candidates answered), question 2 ~52% (11), question 3 ~51% (12), question 4 ~57% (10) and question 5 ~67% (7).

Candidates preferred to address questions with a larger proportion of parts requiring factual answers than question 5, a single part question calling for discussion, criticism, argumentation and evaluation (although all questions contained at least one part of such nature). Several candidates failed to appreciate the need to match content of answer to the associated available mark and thus gave in some cases quite under-developed answers that left little leeway to the marker, while on the contrary setting out much often irrelevant or superfluous detail for parts that carried few marks.

Question-specific detail:

Question 1: Several candidates either misunderstood or did not tackle the notions of binary term-document incidence matrix and inverted index, which are fundamental notions in information retrieval. There were some well-motivated and justified decisions regarding tokenisation issues. The main disadvantage of a binary term-document incidence matrix escaped several candidates. There is more than one reason to keep document frequency information for a term in an inverted index: several suggested only one. There were also several misunderstandings of the term "document frequency". The part on the syntactic vs. semantic web was well answered by some, bringing in evidence of own reading. Others omitted this part, indicating a potential lack of understanding of the focus of the course unit, or perhaps an aversion to discussion-type answers. No matter the reason, 7 marks were lost by such candidates. However, even a minimal attempt may have garnered a few marks, thus it is always worthwhile to attempt an answer rather than leave especially a discussion-type question blank (there being typically no one correct answer for such questions).

Question 2: 5 of the candidates who attempted this question achieved marks under 10/20, with one of these not reaching 5/20, thus the question proved a good distinguisher. The effect of stemming on precision and recall revealed misconceptions in relation to some candidates, possibly reflecting a deeper misunderstanding of the notions of precision and recall. The part on tf-idf revealed some basic misunderstandings of idf, however there were numerous good answers here. Understanding of cosine similarity scores was generally good although some answers addressed only one way that this measure can be used with the vector space model. The discussion part was handled well by some. At least one candidate did not answer this part.

Question 3: Scores were again largely bipolar, with half the candidates who answered this question not achieving 10/20. There were 3 answers in the first class range. Most candidates labelled the diagram correctly, however several indicated a lack of knowledge of the fundamental notions of TP, FP, TN and FN. There were several good answers on precision, recall, F-measure, interpolated precision and precision-at-k, with others either omitting these parts or giving poor answers. Practice in doing the exercises in the course textbook would have helped here. There were again some good answers on phrase and positional indexing, although a few candidates evidently did not possess the relevant knowledge of these notions. The same can be said for the part on text mining.

Question 4: This question was in general well-answered. The part on mapping two fragments of XML to RDF proved a useful distinguisher as several candidates revealed lack of knowledge of one of the basic advantages of RDF. The part on semantic web languages allowed some candidates to demonstrate the scope of their reading and understanding, to achieve a good mark, however others offered a more regurgitative account. The part on reuse of an OWL ontology was also a good distinguisher as several candidates did not properly address disadvantages/advantages and also did not show understanding of how an ontology
can be used to drive a user interface for query/search purposes. For the last part on NER, there were some good answers, demonstrating good understanding, reading around and also familiarity with several of the tools that had been discussed or demonstrated. Some candidates did not maximise their possible marks as they did not address the question, e.g., they did not provide a recommendation.

Question 5 was answered well by all candidates except one, and answers showed in the main good evidence of familiarity with the literature, own reading, knowledge of search engines, applications and scenarios not mentioned in lectures, and ability to structure and carry through an argument. Deep learning was evident through insightful discussions on the future of semantic search with candidates taking up a variety of positions that were generally well justified, exemplified and referenced.

I didn't feel that the course itself went as well as usual. I have no idea why this was. It's always been a course where there are two distinct sets of students--ones who have done the 2nd year Symbolic AI course and ones on ex-SoI degrees who have less formal background, but I've always managed to cope with it before. The only difference this year was that the ex-SoI students were in a minority, whereas previously they've been the majority, and I may have gone too fast at the beginning because I was getting pushed along a bit by the ones who had done Symbolic AI. There was also a weird problem right at the start of the course because no-one could find the lecture theatre--I think some students dropped out right there and then, and the ex-SoI ones mainly didn't find it till week 2, which got us off to a dodgy start. But I put a huge amount of effort into trying to keep the show on the road, and in the end I think (!) that the ex-SoI students who stuck at it got a reasonable grip. It might, however, just be worth looking to see whether their marks on this module are out of line with their other marks.

I can't see any major problems with the exam itself. Every question was done by at least some students, and every question got a good answer from at least one student, so I am comfortable that the exam was fair & that I didn't give them grossly misleading clues in the revision sessions. Some of the questions didn't actually require much writing in order to get full marks, but none of them were trivial (because, even ignoring the one student whose paper indicated that they had barely attended the lectures or read the notes, every question caused at least some students problems). There's a reasonable spread of marks, with some people getting good overall marks and some getting less good overall marks and almost everyone getting a pass mark or close to it. I'm happy enough.

Addendum

In my notes about how COMP37412 went, I expressed concern about the fact that this course had not gone as well this year as in previous years, and about the fact that I had had more difficulty than previously in managing the mixture of students on ex-SoI degrees and ones who had done the second year Symbolic AI course. Having seen the ex-SoI exam results, it seems that the ex-SoI students on this course were in fact someone on an ordinary degree and two of the bottom three students on the honours degree. So it may just have been that these were weak students.