Clinical coders and decision making

Andrew Wooding

Abstract

Clinical coders operate at six identifiable levels, which can be described as beginner, trainee coder, entry, competent, accredited and advanced levels. In this article these levels are elaborated within the theoretical perspectives of Simon’s four-stage and Wilson and Walsh’s six-stage models of systematic decision making. The article then examines briefly the importance of understanding how clinical coders make decisions, because of both the coder’s crucial role in determining hospital funding, and the reliance of research upon access to accurate data. Finally, future avenues for study in this area are suggested.

Keywords: Decision making; clinical coding; health classification.

Disclaimer: This work is the author’s own opinion and does not necessarily represent the views of the Auckland District Health Board.

Introduction

How do clinical coders decide what to code and which code or codes to use? Is there a systematic decision making process that clinical coders use? How can managers ensure that the decision making processes are accurate and effective, thereby resulting in clinically meaningful data?

To answer these questions, this article will first consider the classic four-phase model of systematic decision making derived from management science (Simon 1977) and compare it with a six-phase model proposed by Wilson and Walsh (1996) from a literature review of information-seeking behaviour. Secondly, it will look at the informational and knowledge needs of the six levels through which a clinical coder progresses (Wissmann 1999) and how each level uses the six-phase model. Thirdly, it will explore the issues associated with accurate decision making by clinical coders; and finally, it will propose further avenues of research in this area.

There has been very little work undertaken to investigate how clinical coders think and make decisions. Consequently the second part of this inquiry, the six levels of clinical coding, is of necessity descriptive of the author’s experience in training clinical coders throughout New Zealand (NZ) and Singapore. However, there are obvious parallels here with the various studies on medical decision making in the literature (eg, Patel et al. 1984; Friedman et al. 2001; Allen et al. 1998; Eddy 1990).

Models of systematic decision making

Simon’s model of systematic decision making in management has become a de facto standard throughout the literature on decision making and decision support. It is made up of four phases through which every decision must move:

- intelligence
- design
- choice
- review.

In other words data are presented, formatted as information, a choice is made and the results are examined (Simon 1977). By omitting the review phase from their list of major phases in their chapter on modelling decision support systems in Decision Support Systems and Intelligent Systems, Turban and Aronson (2001) imply that they view this phase to be of minor importance. However, omitting the review phase would leave one without the knowledge that the process resulted in a ‘correct’ decision (or not).

Wilson and Walsh (1996) synthesise several authors’ models and add two more phases to Simon’s:

- intention
- information extraction and integration

Wilson and Walsh suggest that Simon’s focus on machine operations is the reason for his omission of the information extraction and integration phase.

Definitions of the six phases will assist understanding of the process of decision making. By ‘intelligence’ Simon (1977) intends the military meaning of ‘searching the environment for conditions calling for decision’ (p. 40); or, in Turban and Aronson’s words, ‘reality is examined and the problem is identified and defined’ (p. 41).

Norman (1984) defines intention as the ‘mental characterisation of the desired goal’ (cited in Chapter 5 of Wilson & Walsh 1996). In other words, having identified and defined the problem a desired goal is defined.

‘Design’ is defined by Simon (1977) as ‘inventing, developing and analysing possible courses of action’ (p. 41). Wilson and Walsh see this phase as overlapping with the intention phase but it is treated here as a distinct phase.

Simon defines ‘choice’ as ‘selecting a particular course of action from those available’ (Simon 1977: 41).

‘Information extraction and integration’ comes from a synthesis by Armbruster and Armstrong (1993) of models of the knowledge acquisition process by Guthrie and Mosenthal (1987) and Dreher (1992). The concept captured here is the need to integrate extracted information with prior knowledge. This would enable future similar decisions to be made more easily.

The ‘review’ phase Simon defines as ‘the [assessment] of past choices as part of a repeating cycle that leads again to new decisions’ (Simon 1977: 41). Arm-
bruster and Armstrong characterise this phase as ‘re-cycling with monitoring and evaluation of progress towards the goal’ (cited in Chapter 5 of Wilson & Walsh 1996). Norman regards it as ‘a review of the executed action to direct further activity’ (cited in Chapter 5 of Wilson & Walsh 1996).

To summarise, a systematised decision is made through finding the problem, resolving to solve it, examining possible solutions, choosing the best solution according to one’s knowledge of the situation, taking the information learnt and integrating it with one’s knowledge, and then looking to see if the solution chosen has resolved the problem.

Turban and Aronson regard Simon’s model as ‘the most concise and yet complete characterisation of rational decision making’ (Turban & Aronson 2001: 41). However, the six-phase synthesis of Wilson and Walsh (1996) more adequately explains the reasoning and decision making process undertaken by clinical coders, both novice and expert, than does the simplified three-phase model. As shall be seen, the information extraction and integration phase is very important to clinical coders’ decision making. The review phase will also be seen to be important as clinical coders develop their knowledge.

### Decision types

Simon’s classification of decisions into programmed and non-programmed (Simon 1977: 45ff.) is a very cogent one for the clinical coding profession. Some of the decisions that must be made are highly programmed; that is, the decision is a matter of following a particular process. In clinical coding, the situation has occurred so often that there is a specific coding standard telling the clinical coder exactly what to do (National Centre for Classification in Health [NCCH] 2000). For example, Standard 0026 states: ‘Where the reason for admission is stated as being for a clinical trial or drug monitoring, the condition of the patient should be assigned as the principal diagnosis. Z-codes are not necessary in these cases’ (p. 12). Other sources of programmed decisions are rulings by the New Zealand Coding Authority and published answers to Frequently Asked Questions (FAQs), both in NZ and Australia.

Some decisions are non-programmed in that, quote Simon, ‘the system has no specific procedures to deal with a situation . . . but must fall back on whatever general capacity it has for intelligent, adaptive, problem-oriented action’ (Simon 1977: 47). For clinical coders this occurs when there is no coding standard or other source as above to cover a particular coding situation. The ClinicalCoder’s Creed (NCCH 1998) states the general principles under which a clinical coder operates.

### The clinical coding process

There are three primary tasks in clinical coding: abstraction, assignment and review. ‘Abstraction’ is here defined as reading the entire record and analysing the information therein to find out what was wrong with the patient, how it happened and what was done about it. ‘Assignment’ occurs when the clinical coder interprets their abstraction in the light of the ICD-10-AM classification and assigns codes. On completion of code assignment the code set is then grouped and assigned a DRG (Diagnosis Related Group). The clinical coder then reviews the DRG to see if it matches what they read. The coder should also review their code set to check for inconsistencies and completion.

Thus two of the six phases fall under one of the three tasks. The task of abstraction covers the intelligence and intention phases (what are the diagnoses associated with this patient and do they need to be coded?). The assignment task takes the design and choice phases (what codes could be used and which in particular is chosen?). Review picks up the final two phases: information extraction and integration and review (what is there to be learnt here and does this code fit?).

### Clinical coders

The levels of proficiency in clinical coding that are considered here are based upon the several tiers in the Clinical Coding Structure as implemented at Auckland District Health Board (Wissmann 1999). People can enter the profession at any of the first three levels, depending on their previous qualifications.

There are aspects other than those covered in the following discussion that are taken into consideration in advancing a clinical coder through the levels, such as communication with clinicians and data analytical skills. These do not, however, impinge directly upon the decision making that clinical coders use in their day-to-day work.

### Beginner level

The beginner has neither knowledge of medical terminology nor of clinical coding, and cannot really be considered to be a clinical coder. The first step towards clinical coding proficiency is to learn medical terminology (including basic anatomy and physiology). The decisions at this level are curtailed because of the directive nature of the courses and texts; however, the six-phase model is relevant as a learning process. These six phases are very similar to those in Accelerated Learning techniques (Rose & Nicholl 1997). (The six steps in Accelerated Learning are Motivating Your Mind; Acquiring the Information; Searching out the Meaning; Triggering the Memory; Exhibiting What You Know; and Reflecting on How You’ve Learnt).

### Trainee coder level

Having successfully completed a recognised course in medical terminology, the trainee clinical coder then commences a course of training in using the clinical classification and the standards surrounding it. In some hospitals, this is an informal on-the-job process. In others, such as Auckland District Hospital Board (DHB), this process is formalised. People may enter the profession at this level if they have already completed some recognised course in medical terminology or can demonstrate a sufficient knowledge based on a clinical background in medicine, nursing or one of the allied health professions.
The informational needs at this level are sourced from senior coders with whom the trainee is working or from the tutors of the coding courses. The coding exercises and discharge summaries that are used as teaching resources are chosen to highlight aspects of the classification. This means that the intelligence phase of decision making concentrates on interpreting the question in the light of the aspect of the classification being studied.

One would hope that the intention phase is to find the right code. The design phase varies at this level depending on the background of the trainee. If the trainee has a clinical background, information is often gathered as if it was going to be applied in a clinical context. However, if the trainee comes from a non-clinical background, there will be a more limited range of options through which to negotiate. Assuming that the course is well designed, the design phase should have resulted in only one final option for the choice phase and information integration should flow automatically as the trainee is led in a structured fashion to build up their knowledge.

Review occurs when the trainee either checks their answers with the sample answers or receives an assessment. ‘What did I do right?’ ‘What did I do wrong?’ ‘Why was it right or wrong?’

Entry level

Successful completion of a clinical coding course gives the new clinical coder a framework on which to build real-life decisions. The time a clinical coder spends at this level depends on their ability to provide accurate and timely data. Like nurse graduates, coding graduates who enter the workforce after graduation lack skills in solving unprecedented clinical problems with multiple foci’ (Oliver et al. 1999: 2).

In the coding course, the trainee saw neatly typed, well-constructed coding exercises. Now, however, the hand-written, badly organised documents that make up clinical documentation have to be analysed for diagnoses and procedures. This analysis, or ‘intelligence’, phase remains a constant throughout all the ensuing levels, but it is performed differently at the various levels. At this level the clinical coder reads everything in the medical record, taking numerous notes as they do so.

Then the ‘intention’ has to be resolved. Which of these notes relate to each other? Which can be ignored? Which are important? The major fault of clinical coders at this level is ‘over-coding’ (ie, coding more diagnoses or procedures than is necessary to represent the episode of care). This comes about because the clinical coder has not yet integrated the classification and the Clinical Coders’ Creed with the ‘real world’ as presented through clinical documentation.

Once the intention phase is entered the clinical or non-clinical background of the clinical coder comes to the fore. The danger for coders with a clinical background is that their coding could be based on their clinical knowledge instead of only that which is documented. This problem arises because of the knowledge scripts (Edwards 1995, 1997) that are carried across from their original clinical context. It is very difficult to ignore prior knowledge. For the coder with a non-clinical background, there is no prior knowledge and there is a tendency to blindly code whatever is found, rather than determining a coherent whole.

The design and choice phases are sometimes the most challenging at the entry level. Every code seems possible and picking one can be a daunting task. At Auckland District Health Board, we help coders at this level in two ways. The first is that entry-level coders focus on one specialty at a time and then slowly add specialties as they increase their knowledge. The second is that an experienced coder checks all work done by an entry-level coder. We ask them to always choose a code and then the feedback given passes to both the information integration and review phases. After receiving the same issue back several times, the entry-level coder learns techniques of checking their own work before passing it over. A further effect is the initial formation of the coder’s knowledge scripts.

Competent level

Many clinical coders reach this level and choose not to (or are not able to) progress further. They continue to develop their coding skills, but never progress to the more integrated stages of decision making that the advanced coder reaches. The reasons for this are multifactorial, but appear to be based on confidence and the learning style of the individual. (For more information on learning styles refer to Chapter 2 of Rose & Nicoll, 1997).

The competent coder has demonstrated an ability to code accurately in a timely fashion and thus makes many decisions without the need to refer them to another coder for agreement. This can lead to a degree of overconfidence, which is balanced by regular audits. At this level, the clinical coder has learnt which documents do not need to be read as part of the abstraction or intelligence phase, and the notes the clinical coder makes are considerably shorter than those of the entry-level coder. Time spent in the intention phase is also shorter, as the knowledge scripts are of greater depth.

The design phase for a competent coder consists of working through the various options in the classification and often coming up with several possibilities. Through discussion with colleagues or working through the classification manuals a choice is made. The preference, as with medical professionals, is discussion with colleagues (Rosenberg & Sackett 1996).

Competent coders are constantly on the lookout for new information that may be integrated into their knowledge scripts. This is particularly so in those who come from a non-clinical background.

Review at this level is in two forms. Firstly, the competent coder is learning to develop the instinct to detect when a code set looks ‘wrong’. Secondly, regular audits confirm accuracy.

Accredited level

An accredited clinical coder has sat a further examination that demonstrates a thorough knowledge of clinical coding using a particular classification across a wide range of patient types and clinical specialties. Whether this level represents an intermediate step between competent and advanced is as yet undetermined, as the national accreditation program is cur-
rently in abeyance due to the changes in the classification over the previous 6 years.

Advanced level

Advanced clinical coders have demonstrated a very high degree of consistent accuracy and have subsumed the Clinical Coders’ Creed to the point where all the aspects of knowledge involved in clinical coding are fully integrated. They will still consult other clinical coders on individual cases, but this is often to support decisions already made. As they analyse the documentation they are performing the intelligence, intention and design phases simultaneously. The notes they make are minimal and usually act more as reminders of what they have read. Often the notes will consist of a list of codes and abbreviations. This is not a memorisation process, but rather signifies increased lateral experience. The advanced clinical coder happily solves problems never seen before, a state described by Oliver et al. (1999: 2) as ‘unprecedented clinical problems with multiple focus’. These clinical coders can perform at similar levels of expertise with classifications that are new to them (Department of Health and Family Services 1997; Grozef et al. 1999).

A further part of the intention phase for advanced coders is that they automatically check for information that they know arises from the information they already have; for example, the diagnosis of hyperparathyroidism on a surgical case will trigger a search for documentation of the cause. This is because the incidence of secondary hyperparathyroidism (particularly renal) is much higher than that of primary.

The choice phase at this level is frequently a simple one based on the integrated knowledge of the classification. New information is integrated readily into the knowledge scripts an advanced coder has already developed.

The review phase, similarly to the competent level, has two aspects. The first is the regular auditing of their work by other coders. The second, the automatic scanning of the list of codes selected for anomalies.

Why do clinical coders need to make accurate decisions?

There are several implications of accurate decisions by clinical coders. The two most important centre on money and research.

The Clinical Coding Service of any hospital is expected to provide data so that the optimal level of funding for each patient event is received. The more accurate a clinical coder, the more sure the hospital can be that it has received the correct funding for an event. There are, of course, other influential factors, including the quality of documentation and the nature of the contracts a hospital has with its funders. However, the first point of responsibility is, in this author’s experience, the coding. The funding issue can be starkly expressed in terms of a coder’s salary. If a coder was not abstract, and therefore not code, 10 pacemaker insertions over the course of a year, then that coder’s salary will have been lost to the hospital.

In addition, the hospital may be fined for under-performing on its contract.

Research is also dependent upon useful data. The Decision Support Unit at Auckland DHB receives several requests every day for data for research purposes, most of it dependent upon the codes assigned by clinical coders. Without accurate data from the Clinical Coding Service much valuable time on the part of the researcher will be wasted and, at worst, the wrong conclusions will be reached, which may then impact on patient treatment.

The next step

The next step in the process of understanding how clinical coders make decisions is to conduct a study to test whether the models presented above are valid. Such a study will be piloted later in 2004 and then, it is hoped, extended in 2005.

The findings from this study will assist the designers of training and quality management programs for clinical coders to determine how best to direct their programs. Both of these areas are of interest to this author.

An interesting part of the study, if it is possible to expand it, will be to compare clinical coders from the different Australian states and territories, New Zealand and Singapore. Does the region in which a clinical coder works make a difference in how they make decisions? Or is it dependent on which training course they undertook?

Another area that would benefit from an understanding of the decision-making process for clinical coders is decision support. The big challenge in decision support is ‘working out what to give people so that each person individually can learn what they need to learn, when they need to learn it, how they want to learn it and where they want to learn it’ (Peers 1999: 4).

A clinical coding decision support system needs to combine knowledge of best practice in all clinical areas with expert knowledge of the classification in use by the clinical coder. Ideally it would involve Artificial Intelligence so that the system itself would learn how to present information to individual clinical coders and be self-auditing.

Conclusion

A clinical classification is an arbitrary structure that is laid over the top of the ‘real world’. In other words, the real world is much more complicated than can be represented by any of the currently available disease classifications, and indeed procedure classifications. Even the procedure classification developed for future use in the United States of America, The International Statistical Classification of Diseases and Related Health Problems, Tenth Revision, Procedure Classification (ICD-10-PC), with its very large number of potential codes (52 x 10^9), still does not permit some of the finer nuances of surgical work to be captured. This necessitates the employment of trained people who can interpret the real world in terms of the classification and the classification in terms of the real world.
Simon’s four-phase model of how the interpretative decisions are made by these people (clinical coders) is limited because of the omission of the important phases of intention and information extraction and integration. The intention phase has been shown to be involved in working out what information needs to be coded and what can be left out. The information extraction and integration phase has been shown to be important for refining the process for future decisions. As Oscar Wilde (1966) once said: ‘It is a very sad thing that nowadays there is so little useless information’ (p. 1203).

References


Andrew Wooding GradDipBus (HlthInfo)
Team Leader – Clinical Coding (Medical, Surgical & Child)
Auckland City Hospital
Auckland District Health Board
Private Bag 92-024
Auckland, New Zealand
Email: awooding@adhb.govt.nz