1st Annual Conference on the Aiming for Excellence in STEM Learning and Teaching
12 – 13 April 2012, Imperial College, London & The Royal Geographical Society.

TEACHING USE CASE MODELLING USING *FLUXX*®

Nik J. Whitehead
School of Applied Computing
Swansea Metropolitan University
Swansea, SA1 6ED
nicola.whitehead@smu.ac.uk
http://www.smu.ac.uk/research/index.php/dr-nik-whitehead

ABSTRACT
Most systems analysis texts introduce use case diagrams by discussing systems of which many students have no practical knowledge - university registry or library systems, or perhaps sales or appointment scheduling systems. This leads to students having to imagine a range of possible tasks that may, or may not, be relevant to the system, with the result that the students may end up learning how such a system works rather than how to create a use case model of such a system.

An alternative approach, based on experiential learning, is to give the students a task and then get them to create a model of the task they have performed. In this case groups of students were given copies of the card game Fluxx® *and asked to first play a few rounds of the game in order to familiarise themselves with the rules, then to develop use case models of the system at various levels of complexity.*

This paper describes the author's experience of using such an approach with first-year university students from a variety of computer science degree lines, provides some examples of student feedback about this method of teaching.

KEYWORDS
Systems analysis education, experiential learning, card games, software engineering education, game-based learning.

1. BACKGROUND
Teaching within a university context is, in many places, still of the 'chalk and talk' variety. In a field such as systems analysis this tends to be talking about a subject, introducing new concepts, exploring an example or two, then presenting students with long passages of text that make requirements analysis and modelling an exercise in advanced technical English comprehension.

Several studies have looked at the use of specially-designed computer simulations (Hainey, Connolly, Stansfield, & Boyle, 2011) or card games (Baker, Navarro, & van der Hoek, 2005) in the teaching of software engineering processes, but these all require significant amounts of preparation and development time before they can be used with students. A faster alternative is to use commercially available games or traditional games with common materials such as decks of playing cards (Curzon, 2000). These all proceed from the definition of learning as a process wherein experience is transformed into knowledge (Kolb, 1984); game-based learning is, by its very nature, a experiential learning approach.

Using games as teaching tools, as well as providing an interesting alternative to the usual text-based materials (its novelty thus increasing the likelihood of it being remembered), also spurs the competitive

side of many computer science students who play games outside of the classroom. This can be a great motivator, which is a major feature required to support individual learning (Biggs, 1999).

2. Teaching Use Case Modelling – the traditional approach

A common approach used in many foundation-level texts is to use an example situation such as a university library or student record system, or the system required to manage patient appointments at a hospital or doctors' surgery (Dennis, Wixom, & Tegarden, 2010). While these systems are good for showing the complexity of a real-world system and for providing examples of the different features of use case modelling that students must learn, they are by necessity artificial in that they are simplified because a) relatively few individuals teaching use case modelling have worked in these environments and so do not know the actual systems involved, and b) different students may have different views of how the systems work, in particular overseas students may have had very different experiences of a medical service to home students in terms of paying for treatment.

As a result of this, it is quite possible for students faced with a problem such as creating a use case diagram for a university library system, to spend most of their time on the problem trying to work out just what sort of things a librarian might need to do in their job. This can lead to some students feeling overwhelmed by the problem and not knowing where to start. One approach to avoiding this problem is to give the students a problem of which few, if any, of them have any experience but where they can gain experience quickly. Simple card and board games can be used for this purpose.

3. Fluxx – a card game with ever-changing rules

Fluxx® is a card game developed Looney Labs and now into its fourth main edition with a number of separate themed versions **(Looney Labs, 1998)**.The game consistes of a deck of cards divided into 5 groups:

KEEPERS – cards showing an object; these cards are, once played, kept in front of the player, e.g. the Moon, Chocolate, or Dreams.

GOALS – cards showing which combination of objects a player has to have in front of him in order to win the game, e.g. Rocket to the Moon, which requires the Rocket and the Moon; or Chocolate Biscuits, which requires Chocolate and Biscuits.

ACTIONS – cards which allow the user to perform an unusual action such as taking an extra turn, taking new cards, or stealing a keeper from another player.

CREEPERS – cards which are like Keepers but prevent the user from winning if they have a Creeper on the table in front of them. Players have various options for removing or redistributing Creepers

NEW RULES – cards showing a new rule which comes into operation immediately that the card is played.

Examples of these Fluxx cards are shown in figure 1 below:

Figure 1 Examples of Fluxx cards

Players are initially dealt three cards, after which the game starts with the dealer and the basic rules of draw 1, play 1 are applied with the current playing drawing a single card and then playing one of the

four cards in his hand. Play continues around the table until a player plays a New Rule card which comes into force immediately. If, for instance, the new rule is *Draw 3* then the player must draw another two cards to make a total of three drawn cards in total.

The game then continues, and the new rules played by the players operate cumulatively so that there may be a large number of rules in play at any one time, e.g. number of cards drawn, number of cards played, number of keepers allowed in front of a player, and number of cards that can be held in the hand. All of these are visible at all times so that players can refer to them. Whenever a goal is played it replaces any goal currently in play. A player wins the game as soon as he meets the criteria of the current goal, no matter whose turn it is at present.

4. USE CASE MODELLING WITH FLUXX

After explaining the underlying concepts behind use case diagrams and working through several examples in a lecture, the students were split into groups of 3-4 and each group was given a copy of one of the various editions of Fluxx. All of the editions have the same core set of rules and actions but each edition has a different themed set of goals, actions and extra rules. The students were given the time to play the game and asked to consider the tasks involved in playing the game.

In its most basic form the use case diagram is simple, as shown in figure 2 below:

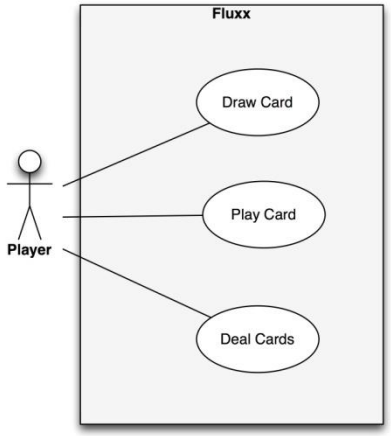

Figure 2 Basic Use Case diagram

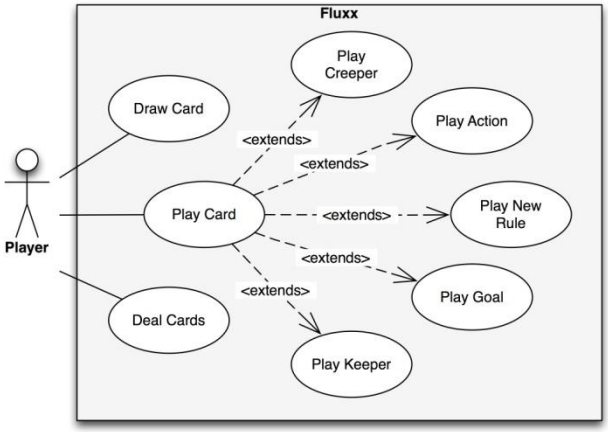

Figure 3 Initial extended use case diagram

Students were able to create this diagram quickly, and then extend it to that shown by figure 3 *even before the concept of <<extends>> was introduced in the next lecture.* Once further concepts had been introduced, the students were able to extend the diagram further, as shown in figure 4.

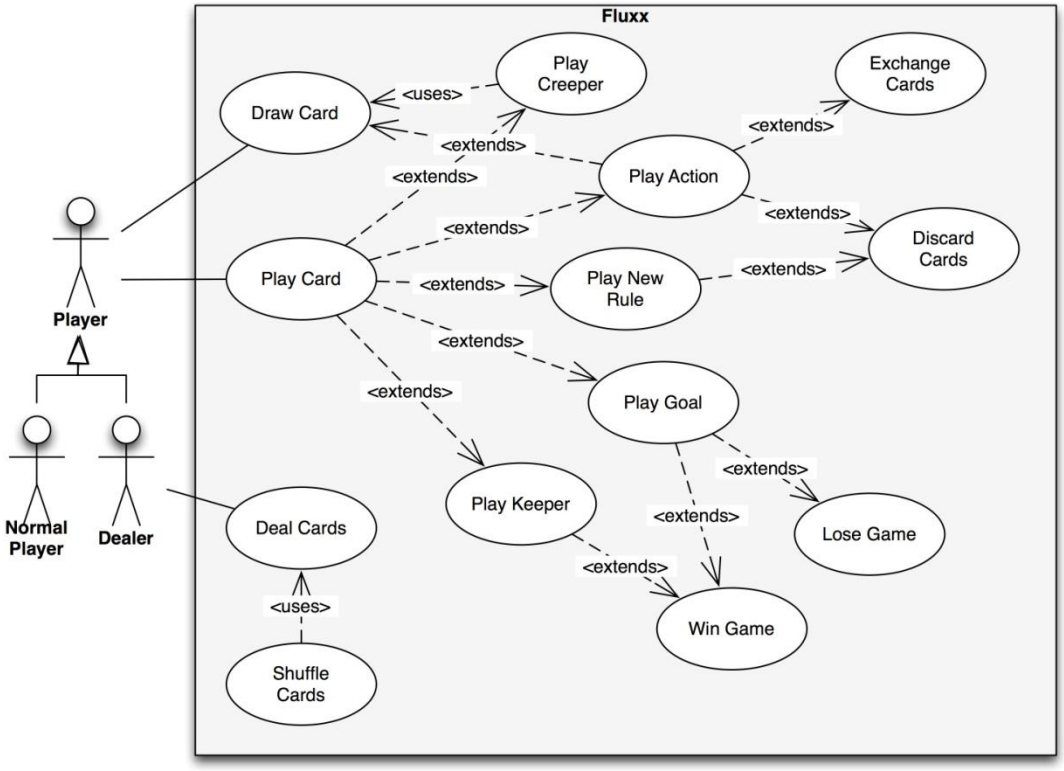

Figure 4 Further extended use case diagram

In fact the use case diagram grows increasingly complex as the results of individual action cards are added to the diagram, which provides an example of how quickly even a relatively simple problem can become so complex that it needs to be split into subdiagrams.

5. STUDENT RESPONSE TO FLUXX

Fluxx was used with approximately 100 students over five degree lines. After the tutorial sessions they were asked about their experience of learning using Fluxx. The student response to using Fluxx was very positive overall, although more positive in some degree lines than in others. Students studying software engineering or web development were most positive, while students studying business information technology were less positive. Within the more positive groups, the idea of using games as learning tools was thought to make learning more interesting even when the modelling problem was viewed as quite difficult:

> *It is a good idea because Fluxx is a novel concept so it is interesting.*

Other students who found use case modelling easier enjoyed the tutorial sessions more than usual:

> *Good idea, playing Games while learning!*

> *I think it's a good idea as you tend to remember fun better.*

All previous tutorial work had been based upon textural descriptions, a method supported by the standard teaching texts. This requires the student to have a good grasp of English in order to extract the important details of the text. One student commented:

Learning the game was a relaxing break from the constant pages of text we're assaulted with, and it took roughly the same amount of time to digest the games rules as it would to understand a system of equal complexity presented in text form.

With the less positive responses it was noted that the groups who were least positive about the use of Fluxx were those who admitted to seldom playing board or card games of any sort. It was clear during the tutorial sessions where Fluxx was used that these students were having problems understanding the game even with the rules in front of them..

6. DISCUSSION

The author expected that a majority of students would prefer learning using a game to the more traditional approach, and this was true with students from the more computer-facing degree lines. The business IT students had considerably more problems, possibly because they are less inclined to play games than the more technical students. As a result of this, it is unlikely that Fluxx will be used with this group in future.

A further issue that arose was due to the high proportion of students with English as a second language in the business IT group. Although all of the students had access to the rules of Fluxx and could use them for reference, those students whose English was weak had greater problems working out the rules. These students often have similar problems when handed a text from which to extract information for modelling, so it is the author's belief that the no real conclusions can be made about the usefulness of Fluxx with these students unless an Arabic copy of the rules (or the entire game) can be produced.

As a result, the author intends to continue using Fluxx with all degree lines other than the business IT group and will investigate the practicality of getting an Arabic translation of the rules for the mainly-Arabic-speaking overseas students.

REFERENCES

Baker, A., Navarro, E. O., & van der Hoek, A. (2005). An experimental card game for teaching software engineerind processes. *The Journal of Systems and Software 75* , 3-16.

Biggs, J. B. (1999). *Teaching for quality learning at university.* Maidenhead, UK: Open University Press.

Curzon, P. (2000). Learning computer science through games and puzzles. *Interfaces 42* , 14.

Dennis, A., Wixom, B. H., & Tegarden, D. (2010). *Systems Analysis and Design with UML - An Object-Oriented Approach.* Hoboken, US: Wiley.

Hainey, T., Connolly, T. M., Stansfield, M., & Boyle, E. A. (2011). Evaluation of a game to teach requirements collection and analysis in software engineering at tertiary education level. *Computers & Education 56* , 21-35.

Kolb, D. A. (1984). *Experiential Learning.* New Jersey: Prentice-Hall Inc.

Looney Labs. (1998). Fluxx. *Card Game , 4th Edition*. College Park, MD, USA: Looney Labs.