Chapter 8. Learning to learn

The learning aim of this chapter is to be able to design activities where people can become better learners of IT.

Information technology is characterised by a quick turnover of new software versions, information systems and hardware gadgets. Users therefore need to upgrade their competence often, meaning they have to learn about the technology and also about the two other subject matter areas described above. This implies that IT user competence also includes the competence for learning about IT, which includes learning about information, technology, and tasks.

This chapter will first summarise the competence that improves the conditions for learning in Section 8.1.

In the previous chapters, we have presented externalisation as the learning process for IT competence. Proficient IT users seem to engage also in other learning processes, like playful exploration of the technology, experimenting with its functionality and finding solutions to problems by searching the internet or calling friends. Most users would need to be introduced to these learning processes in order to engage in them on their own at a later stage. This chapter will therefore present these processes in Section 8.2 and 8.3, such that they can be introduced by the IT teacher.

Since the learning processes described are applicable to all three subject matter areas,

8.1. Competence for learning

Learning is based on what we already know (Bransford, 2000, pp. 10, 15-16, 68-71). When learning a new operation, we adapt our existing competence to the new situation.

When learning a new piece of software, people bring their experience of doing the tasks without computers or with other software. Knowing that text can be duplicated through a copier, they might type their text in the text processor, print it, multiply it in a copier machine and send one copy to each receiver. After having also learnt about electronic mail, they might reconsider this as a way of distributing text.

Novice users learning to use a text processor may have been used to creating one long sequence of text from page to page, and they have read newspapers where the text is divided into columns. If they get the task of organising text into two columns like in the newspaper, they may choose inserting a table with one row and two columns, since this gives the same appearance, even though they manually have to move text from one column to the next one.

Users come with blurred concepts which do not match the principles of the software package, or they have developed more precise understanding from using one piece of software and have trouble when their expectations based on this concept are not met when using another program. For example, users often confuse

- aligning text with the tab key in a text processor
- columns in a text processor
- tables in a text processor
- tables in a spread sheet

A poor learner will not be aware of the possible mismatch between preconceptions and novel technology. On the other side, a person with learning competence in this respect will be open for all options of whether aspects of a concept fit in a new environment or not.

Pedagogical literature also points out other conditions for learning, and understanding usefulness is one of these (Bransford, 2000, p. 61), as also mentioned in the previous chapter.

Understanding and facts ease transfer of skills to new situations (Bransford, 2000, pp. 9, 16-17, 63, 65). A computer user who has understood the concept of text flow, and that text flows from one column to another, but not between cells in a table, would be more likely to choose the right kind of tool in the next piece of software. Understanding the three subject matter areas presented in the previous chapters therefore ease learning of new IT concepts, information and use of IT in tasks.

Managing a new text processor would also be eased if a user has learnt trivial facts like that Times New Roman and Arial are fonts, because then it is easier to recognise that where these words appear, the functionality concern fonts. Facts like vendors, file formats, and knowledge of people to ask for help would constitute parts of the learning competence for computer users.

8.2. Five basic ways of learning IT use

The three previous chapters have described learning as a process of externalisation from skills to understanding, and this is one of the basic learning processes. Since obtaining the practical experience seems necessary for most people to learn abstract topics, externalisation is fundamental for learning IT. However, there are also learning processes for skills alone, from understanding to skills, for developing theoretical understanding, and for applying the understanding in practice. Some of these are employed by advanced IT users, implying that learning these ways of learning is advantageous. The simpler ones are presented in this section, while blends are discussed in Section 8.3.

Imitation

A basic way of learning is to imitate others' behaviour. For example, Maria observing the colleague uses a keystroke shortcut instead of moving the mouse to hit a button, and that this seems to be more efficient. Therefore she starts using the same keystrokes herself. In a behaviouristic learning perspective, this way of learning is called ‘modelling,’ (Ormrod, 1995, p. 141), but since the word ‘model’ is used a lot for other purposes within IT, we use the term imitation for this learning process. When Maria imitates her colleague, she increases her repertoire of skills, so imitation is a process from others’ skills to ones own, as illustrated in Figure 34.

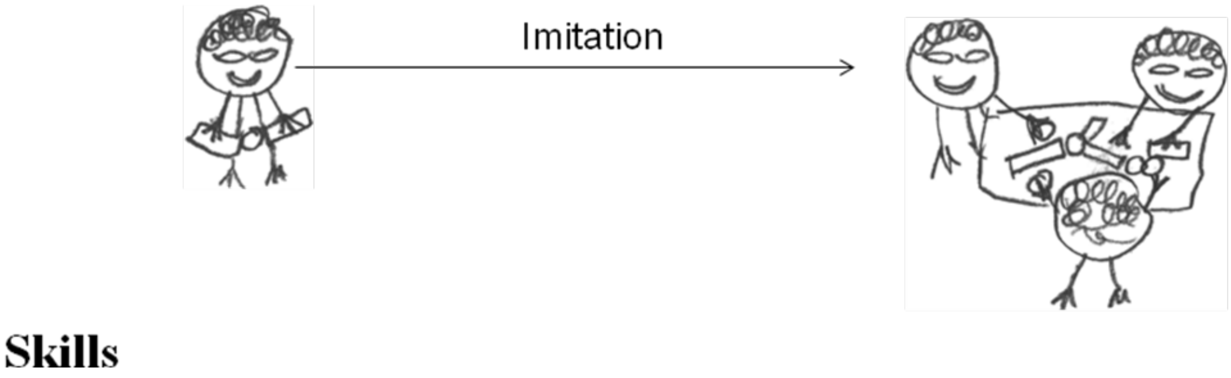

Figure 34. Imitation is learning skills through observing others doing it.

Imitation is a basic way of learning, which also animals employ. It is also useful for learning skills in abstract domains, for example, observing others solving a mathematical problem improves the observers’ success.

Getting users to employ imitation should be a simple challenge for the teacher. In the conceptual-practical training, the instructor could, for instance, demonstrate a shortcut or another option in the summary section. The course participants could learn two things from such a brief session; the particular skill and that imitation is a way lf learning IT.

One may wonder, if imitation is so easy, why is the parrot style of repeating the instructor at the projector such a bad idea? First, users following an instruction sheet is also a form of imitation. Second, this is a better form, because the learner can repeat it after the course while the teacher is unavailable. Third, learning a long sequence of details require a long time, often longer than is possible in a course. Again, the instruction sheet enables continued imitation after the course.

Exploration

Kids have fun with exploring new devices, and they play together and discover. Nerds do the same with IT, developing skills at high level, and they are learning oriented in the IT domain. The learning oriented users actively explore the technology, trying to find better ways of using a program for a certain task, and play around with it in order to see what it can do. This is a strong learning competence. The active explorers have a tendency to become local champion, whom others ask for help and who push for new computer applications.

An example from a study of user competence, Youssou says:

My brother is truly amazing. For myself, if something doesn't work I might try it again once but the majority of the time I will just 'give up'. My brother sees these 'failures' as challenges to be met and conquered. He delights in the fact that he never has to stop learning because there will always be a new challenge to conquer. He loves the fact the information technology is such a dynamic field that it is always changing, improving and making new breakthroughs. (Phelps, Ellis, & Hase, 2001)

The performance oriented users stick with one way of using a program when they have learnt that way, even though there might be easier ways. They refrain from pushing a button which they have not touched before, due to being anxious for making a mistake of loosing data. The anxiety can be regarded as a negative computer learning competence.

(Phelps, et al., 2001) provides an example of this type of learner too, we call her Ofra:

If something goes wrong when I am using the computer I freak out and panic, but when I see these people use the computer they seem to be able to work it out on their own. It is obvious to me that I learn differently to them when it comes to information technology.

A person may be performance oriented in one aspect of life, while learning oriented in another. The stereotypical image of a computer nerd is that he has learnt everything about the computer, but socially, he sticks to what he knows, which is chatting with other nerds. Likewise, the elderly social worker is fabulous in dealing with people, but she has computer paranoia.

Exploring IT means trying it out, being open to the tasks that it can support, and without having particular hypotheses about its function. Exploration is not directed at solving a particular problem, neither in that IT or an external task, it is rather a playful interaction.

Like imitation, exploration extends the repertoire of skills, but through individual or group interaction with the technology, and not through observing masters. The basic learning mechanism is also similar to imitation, in that the learner observes that some actions yield more positive outcome, either by observing the IT or the co-explorers.

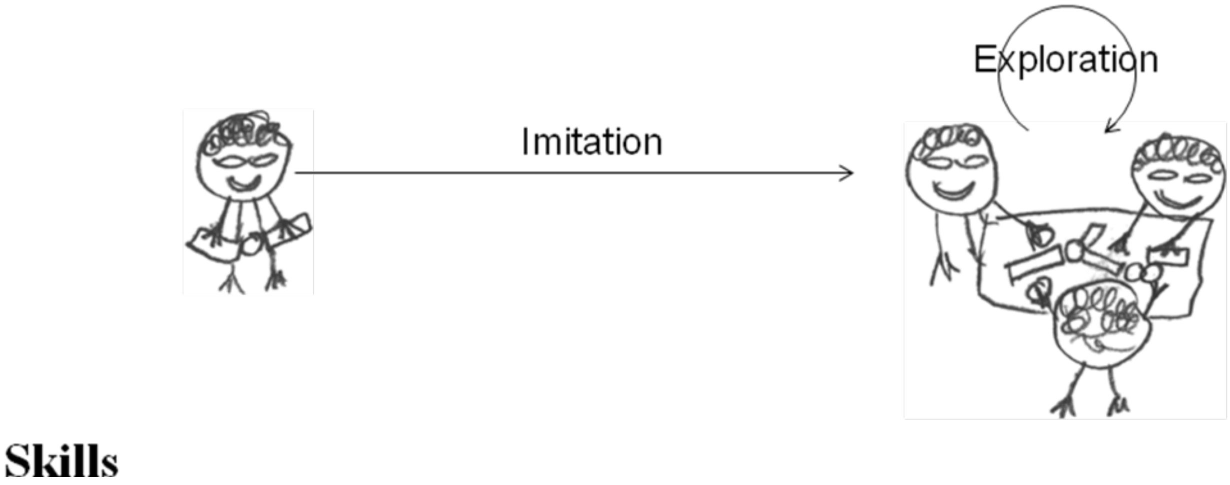

Figure 35. Exploration is learning skills through observing the results of actions.

Knowing that kids explore their environment and their toys by playing, and that animals also learn through playing, one could assume that there is no need for teaching people the process of exploring. Sadly, this assumption is wrong, as the case of Ofra illustrates.

Performance oriented learners like Ofra need to feel secure. The trainer could try convincing them that when things do not work, it is not because they have destroyed the computer. Also, reminding them that there is normally an undo operation which can bring them back to where they were could calm their nerves.

Second, a setting that encourages exploration needs to be created. In a course, the trainer could try triggering exploration by prompts like

- Find out what this program does. The first reasonable response is awarded.
- First row: left button. Second row: right button. After five minutes, you explain to the other row what the buttons do.
- Now, we will try to make this program do something that it was not designed for. Use the spreadsheet application to draw a face.

Some IT lends itself easier to exploration, for instance the Lego robots, while an enterprise resource planning system seems to have control as its main purpose. An option is exploring its applicability to other tasks than its purpose, like the spreadsheet for drawing a face. The teacher could e.g. say

- Find other tasks where you could apply this application

A presentation program could be the best alternative for novices when starting learning IT. It allows for exploration after a quick introduction, users can write and make illustrations, it has built in design elements to be composed, and users can include their own pictures. Kids also find excitement in presenting what they have made, and there is no wrong answer (Det mangler en referanse til Bostrøm her). It would possibly work equally well for adults.

Experiencing exploration constitute one step for users to adopt it as a learning activity. Another step would be to discuss exploration as a way of learning, so that users also externalise learning strategies for IT. The teacher could trigger discussions on what users learnt during an exploration, and point to specific activities which seemed particularly productive in the class.

Combination

Externalisation takes us from skills to understanding. Combination is a cognitive or social process where some elements of understanding are combined into new ones. Similar terms are theoretical learning, abstract learning or conceptualisation.

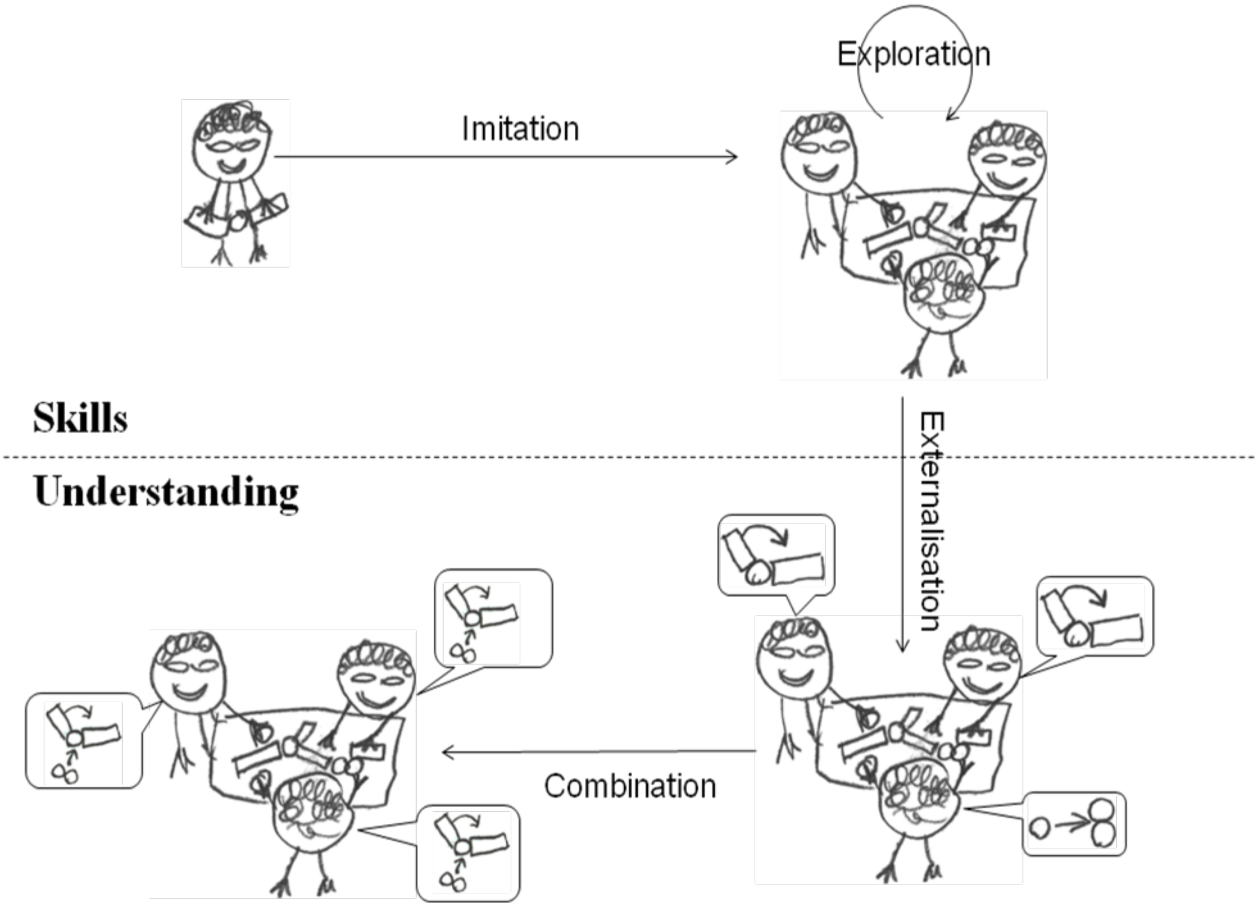

Figure 36. Combination is transforming understanding into new understanding.

An example is that we have understood import by link, and previously we have understood formulas in spreadsheets. When combining, we see the similarity of the two, that when data is changed, the changes propagate. If our previous understanding of the IT concepts was at a functional level, the comparison will contribute to bringing it up to the conceptual. As a second step, when being introduced to the more general concept of functional dependency, we can also combine that with the two former ideas.

When combining, we operate with explicit knowledge or know-that (Section 5.2), which might have been formed through externalisation, see Figure 36. Talking, reading, writing, illustrating or other means of representing concrete experience is needed for combination. During exploration, it is likely that some learners will combine spontaneously and discuss with the group. The teacher has the obligation to introduce novel concepts and link them to the participants' understanding.

Trainers who have lead the class through skill exercises may mistakenly believe that the learners have reached a level of understanding of the concepts. Combination may fail if the learners only are at the skill level.

In order to succeed with combination outside the classroom, users need ways of finding other understanding which can be combined with theirs. One way of doing this is discussing with peers or super-users, so the course trainer needs to bring up this issue and making the users aware of those who are supposed to be more competent.

The other way is searching for written explanations, either on the web or in the documentation following the program or device. This is a process with many pitfalls. The terms known by the users may not correspond to those found in the documentation, misleading the search. Also, there may be large numbers of search hits, many irrelevant, making it difficult finding the right ones.

Practicing searches in class would be the teachers' response, so that the users can be guided in the process of selecting the relevant hits and interpreting explanations.

Internalisation

This process concerns applying theoretical understanding for developing skills. Assume that Nana has learnt ideas of formatting, based on experience in a presentation program, and thereafter she gets introduced to a text editor. A useful learning strategy is trying to apply the same operations there. If successful, Nana will extend her repertoire of skills.

In order to trigger internalisation during courses, the trainer could ask the participants about functionality they know from other applications and request that they try it out in the new one as well. Again, this is a learning process which advanced users would engage in, so pointing to it during training could help users also apply it on their own.

Help can also be found in the built in guidance in software and on the web. Having reached an understanding of formatting, Nana might search the web for "how to format text" and get 88 million hits. Some will provide instruction sheets, which will ease the internalisation.

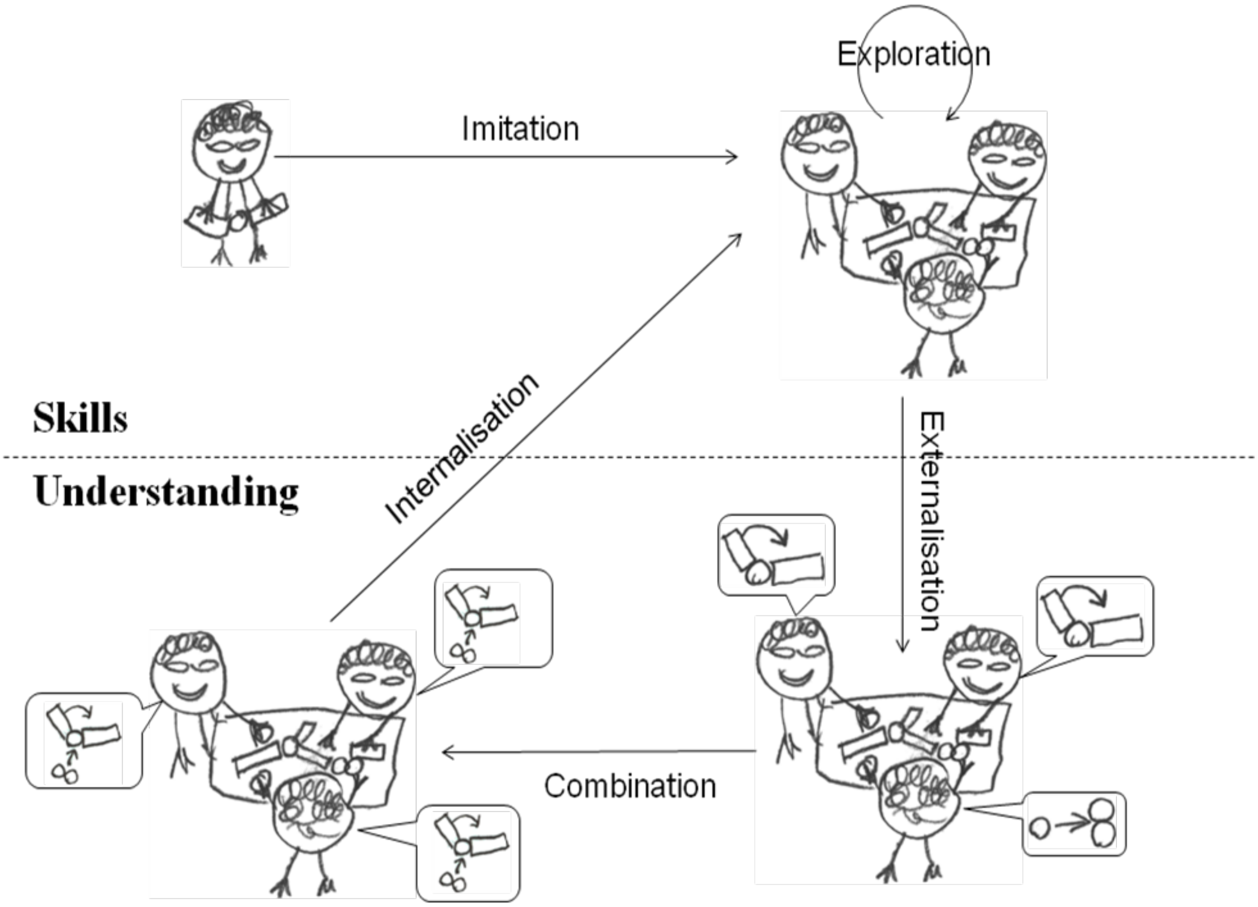

Figure 37. Internalisation is applying previously learnt ideas in new settings.

8.3. Blended learning processes

The categorisation of basic learning processes are organised according to the Socialisation, Externalisation, Combination and Internalisation model of knowledge development in organisations by (Nonaka, 1994). Two distinct processes of socialisation were identified in IT use, imitation and exploration. This section will present how these basic processes are blended in other typical IT learning situations.

Experimentation

Exploration is learning new skills through playing around with the IT.

Experimentation is a planned action, starting with a combination of the type “It can do this. I wonder whether it also can do that. How can we test whether it can? Possibly by doing so and so.” Then the test is carried out, and the result is compared to the initial question.

Advanced users experiment a lot. They test the limits of software, for example “This field is for numbers. Will it take text also?” or “Can the scanner transfer data directly to the phone, or do I have to use the computer as a receiver?” Experimentation includes combination, internalisation, exploration and externalisation, and it provides understanding, as illustrated in Figure 38.

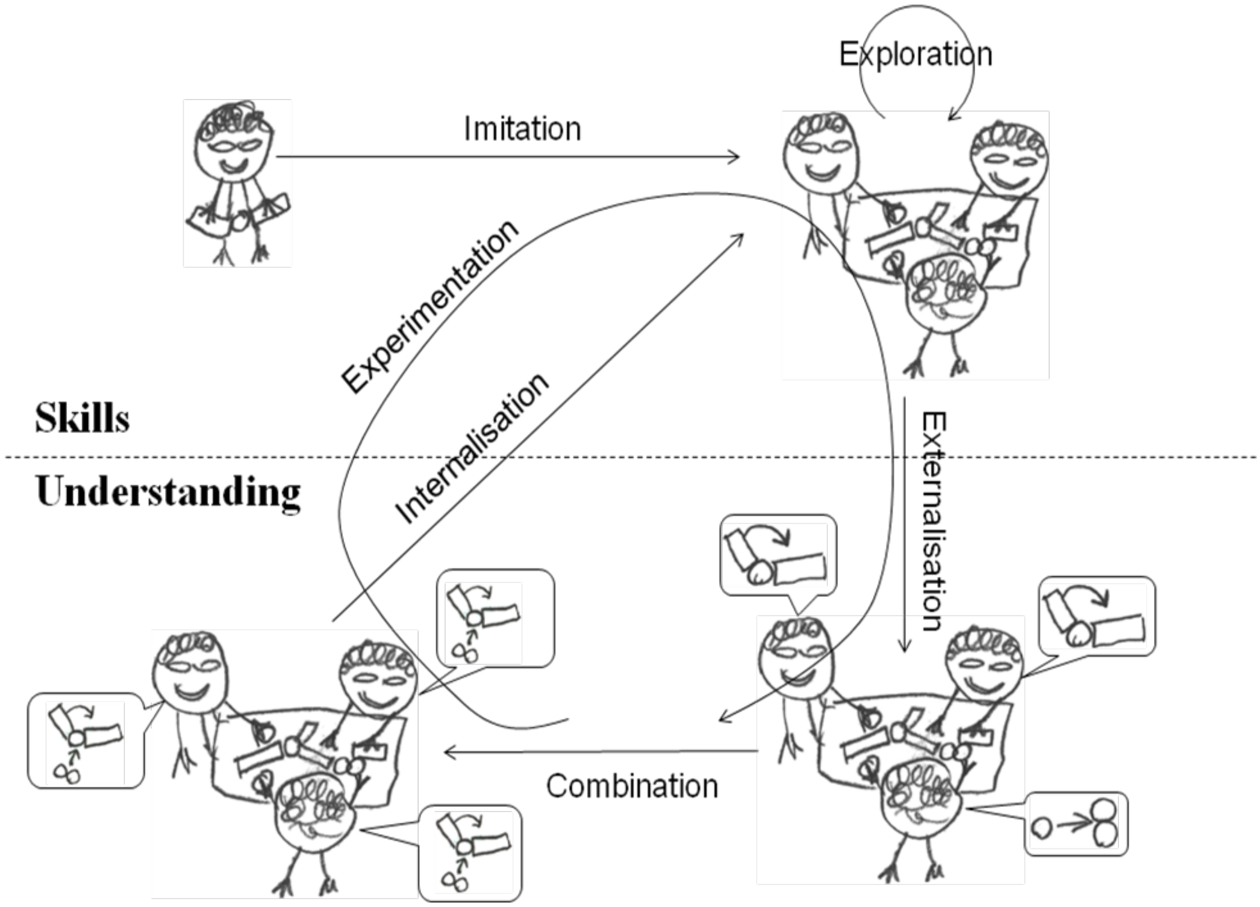

Figure 38. Experimentation is composed by the basic learning processes, and it provides understanding.

A trainer might try triggering exploration by saying “Find out what this program is doing!” The learners may well end up in experimentation nevertheless, discussing hypotheses about what the program can do, testing them, and comparing the results with the hypothesis. Such a systematic approach to learning has the same components as the standard hypothetico-deductive research method. Trainers could encourage it by questions like

- Recall that in the presentation program, we could control the layout of slides with the master slide. How does the text processor compare on this functionality?
- This is a task which you might encounter at work. Find out whether the IT can support this!

The latter request brings the course activities closer to challenges to be encountered after training. It is also an example of problem based learning where the teacher has described a reasonably realistic situation which the learners recognise. When the challenge is complex, the learners will have to go through many cycles of experimentation. For efficient learning, the trainer needs to provide lots of support during such problem based exercises (Hmelo-Silver, et al., 2007).

Trouble shooting

Experimentation was triggered from understanding. Trouble shooting is based in practice, when noticing that the IT did not do as hoped for. We might ask a colleague or search the web, and in both cases we externalise the issue and bring it into the verbal world. Through combinations as described in Section 8.2, we might find a possible solution, which has to be tested. Trouble shooting has the same ingredients as experimentation, but its end points differ, see Figure 39.

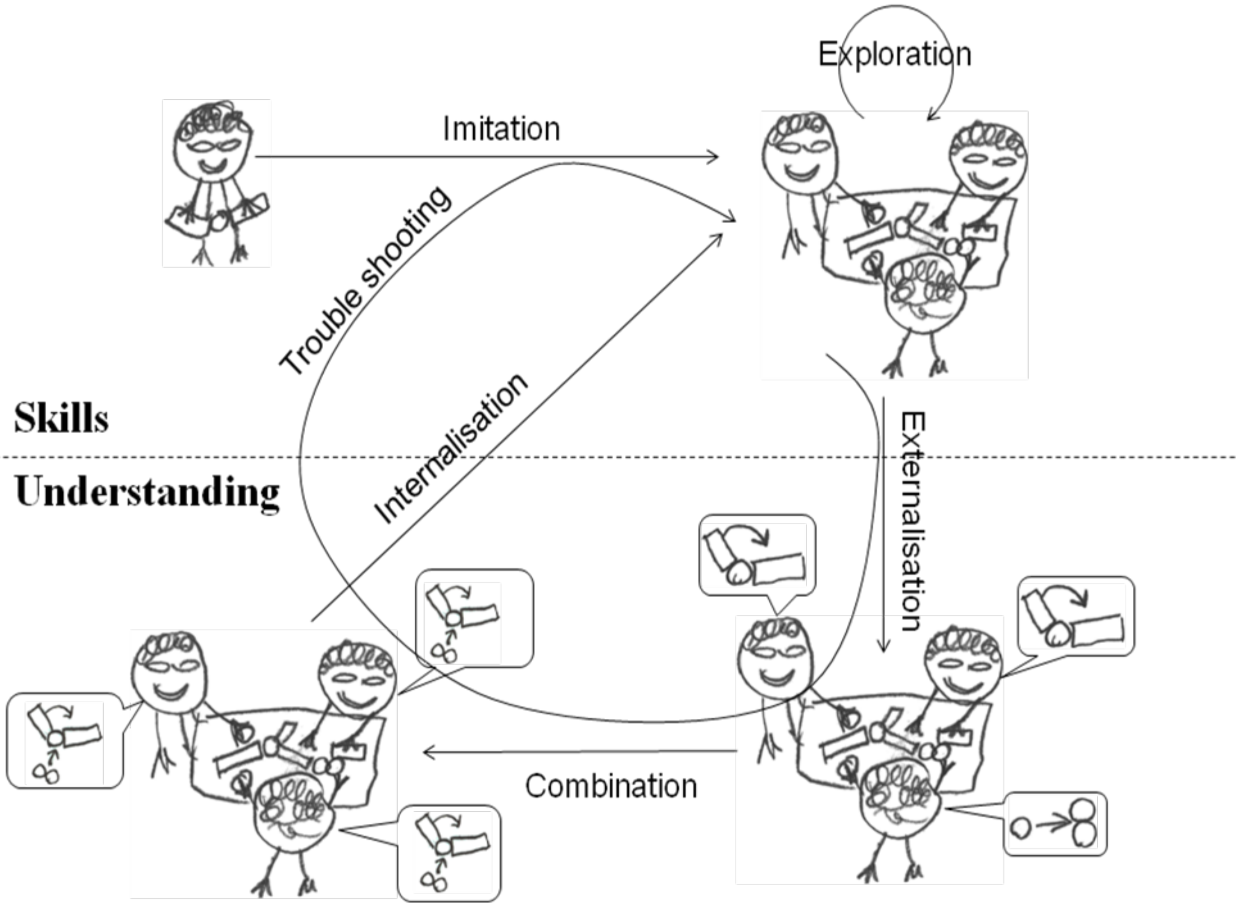

Figure 39. Trouble shooting is composed by the basic learning processes, and it provides skills.

Computer users experience trouble quite often, and again, learning oriented people have better chances of carrying out the trouble shooting loop, possibly including discussions with others during combination. Performance oriented people may also get their trouble fixed, but chances are that others do the job to a large extent.

In any case, finding the solution is no guarantee for learning at all. The problem may appear in a slightly different way the next time, so that the externalisation of it into a known category does not work. If externalised, the user may not remember where to find the information needed and how to search for it. Also, even if the user understands the issue, they way of implementing it in the system might have been forgotten.

Our forgetfulness cannot be mended in general, but learning the steps of trouble shooting procedures could bring performance oriented users closer to the learning oriented ones. During externalisation, the opportunities for misunderstanding are ample.

This worked last time I did it, why did the computer do something else now?

says Elīna. Simultaneously, the trainer has noticed that this time she hit another button than the last time, while the she is convinced that she repeated exactly the same typing. The trainer's response could be to ask Elīna starting over again and re-typing. If the computer performs as she expects this time, the trainer could bring up the issue of observing precisely what one is doing.

Another failure in the externalisation stage is avoiding observing the computer response precisely. The support personnel will often ask the user for the contents of error messages, and if these are lost, reproducing the situation is difficult. Therefore, taking complete and precise notes of what is happening when errors occur should be taught during IT training. This includes saving screen-shots and intermediate results.

Some advices for the teacher were provided in Section 8.2, and these are also valid during combination in trouble-shooting. Also, the teacher can bring up Frequently Asked Questions lists from the technology taught and the learners can try searching for it on the web. A way of getting help when an error message appears is to type the message in a search engine and try reading a hit. The "404 error" gives 8 million hits and the first one is from Wikipedia:

*The **404** or **Not Found** error message is a HTTP standard response code indicating that the client was able to communicate with the server, but the server could not find what was requested. 404 errors should not be confused with "server not found" or similar errors, in which a connection to the destination server could not be made at all. A 404 error indicates that the requested resource may be available again in the future.* (HTTP 404)

In order to understand this explanation, the user should already have understood the concepts client and server. Further, if not knowing what "HTTP standard response code" is, the user should understand that this can be ignored. There might be other hits which fits the novice user better and still others intended for the advanced ones.

8.4. Hole in the wall – the Indian experiment

Se (Mitra et al., 2005).

8.5. Summary

Users who are better able to learn on their own become more proficient and require less attention for support services. Improved understanding increases the chances of learning on your own. Explicit training in the processes of exploration, experimentation and trouble shooting would improve users' learning abilities.

5. Train users so that they can learn on their own.

References

Aharoni, D. (2000). Cogito, ergo sum! Cognitive processes of students dealing with data structures. *SIGCSE Bulletin*, *32*(1), 26-30.

American Library Association and Association of College and Research Libraries. (2000). Information Literacy Competency Standards for Higher Education. Chicago: Association of College and Research Libraries.

Bloom, B., Englehart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. (1956). *The Taxonomy of Educational Objectives, The Classification of Educational Goals, Handbook I: Cognitive Domain*. New York: David McKay.

Boudreau, M.-C., & Robey, D. (2005). Enacting Integrated Information Technology: A Human Agency Perspective. *Organization Science, 16*(1), 3-18.

Bransford, J. (2000). *How people learn: brain, mind, experience, and school*. Washington, D.C.: National Academy Press.

Buschman, J. (2009). Information literacy, "new" literacies, and literacy. *Library Quarterly, 79*(1), 95-118.

Committee on Information Technology Literacy. (1999). *Being Fluent with Information Technology*. Washington, D.C.: National Academy Press.

Coulson, T., Shayo, C., Olfman, L., & Rohm, C. E. T. (2003). *ERP training strategies: conceptual training and the formation of accurate mental models*. Paper presented at the Proceedings of the 2003 SIGMIS conference on Computer personnel research: Freedom in Philadelphia--leveraging differences and diversity in the IT workforce, Philadelphia, Pennsylvania.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly, 13*(3), 319-340.

Dreyfus, H. L., & Dreyfus, S. E. (1986). *Mind over Machine: The Power of Human Intuition and Expertise in the Era of the Computer*. New York: The Free Press.

Duarte, N. (2008). *slide:ology: The Art and Science of Creating Great Presentations*. Beijing: O'Reilly.

Grudin, J. (1994). Social dynamics: eight challenges for developers. *Communications of the ACM, 37*(1), 93-104.

Herskin, B. (2006). *Brugeruddannelse i praksis*. Copenhagen: Nyt Teknisk Forlag.

Hmelo-Silver, C. E., Duncan, R. G., & Chinn, C. A. (2007). Scaffolding and Achievement in Problem-Based and Inquiry Learning: A Response to Kirschner, Sweller, and Clark (2006). *Educational Psychologist, 42*(2), 99-107.

. HTTP 404. (6 March 2011). *Wikipedia* Retrieved 8 March, 2011, from http://en.wikipedia.org/wiki/HTTP_404

. John von Neumann. (2010) *Wikipedia*.

Lankshear, C., & Knobel, M. (2008). *Digital Literacies: Concepts, Policies and Practices*. New York: Peter Lang Publishing.

Leu Jr., D. J., Kinzer, C. K., Coiro, J. L., & Cammack, D. W. (2004). Toward a Theory of New Literacies Emerging From the Internet and Other Information and Communication Technologies. In R. B. Ruddell & N. J. Unrau (Eds.), *Theoretical Models and Processes of Reading. Fifth Edition* (pp. 1570-1613). Newark: International Reading Association.

Lim, K. H., Ward, L. M., & Benbasat, I. (1997). An Empirical Study of Computer System Learning: Comparison of Co-Discovery and Self-Discovery Methods. *Information Systems Research, 8*(3), 254-272.

Luehrman, A. (1980). Should the Computer Teach the Student, or Vice-Versa? . In R. Taylor (Ed.), *The Computer in School: Tutor, Tool, Tutee* New York: Teachers College Press.

Marcolin, B. L., Compeau, D. R., Munro, M. C., & Huff, S. L. (2000). Assessing User Competence: Conceptualiation and Measurement. *Information Systems Research, 11*(1), 37-60.

McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. *Communications of the ACM, 49*(8), 90-95.

Mitra, S., Dangwal, R., Chatterjee, S., Jha, S., Bisht, R. S., & Kapur, P. (2005). Acquisition of computing literacy on shared public computers: children and the "hole in the wall." *Australasian Journal of Educational Technology 21*(3), 407-426.

Nonaka, I. (1994). A Dynamic Theory of Organisational Knowledge Creation. *Organization Science, 5*(1), 14-37.

Ormrod, J. E. (1995). *Human Learning*. Englewood Cliffs, New Jersey: Merrill.

Phelps, R., Ellis, A., & Hase, S. (2001). The role of metacognitive and reflective learning processes in developing capable computer users. *Meeting at the crossroads: proceedings of the 18th Annual Conference of ASCILITE*. Melbourne: Southern Cross University.

Puri, S. (2007). Integrating Scientific With Indigenous Knowledge: Constructing Knowledge Alliances For Land Management In India. *MIS Quarterly 31*(2), 25.

Reynolds, G. (2010). *Presentation zen design : simple design principles and techniques to enhance your presentations*. Berkeley: New Riders.

Sein, M. K., Bostrom, R. P., & Olfman, L. (1998). Conceptualizing IT training for the workforce of the future. *SIGCPR, 30*(1), 223-241.

Sharma, R., & Yetton, P. (2007). The contingent effects of training, technical complexity, and task interdependence on successful information systems implementation. *MIS Quarterly, 31*(2), 219-238.

Stamatova, E., & Kaasbøll, J. J. (2007). Users' Learning of Principles of Computer Operations. *Issues in Informing Science and Information Technology, 4*, 291-306.

Stodolsky, S. (1988). *The subject matters : classroom activity in math and social studies*. Chicago: University of Chicago Press.

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly, 27*(3), 425-478.

Walker, C. B. F. (1987). *Cuneiform*. London: British Museum Press.

Wood, H., & Wood, D. (1999). Help seeking, learning and contingent tutoring. *Computers & Education, 33*, 153-169.

Yi, M. Y., & Davis, F. D. (2003). Developing and Validating an Observational Learning Model of Computer Software Training and Skill Acquisition. *Information Systems Research, 14*(2), 146-169.