Chapter 1

Introduction

Data about students and their learning process is recorded by electronic learning environments. Learning management systems (LMSs), such as Blackboard and Moodle, register grades on assignments, examination results and, for many courses, the resources learners have visited. Interactive learning environments for specific domains, for example cognitive tutors or inquiry-based simulation and modelling tools, record all actions learners perform and the products they create. In this way learning generates large amounts of data that can form the basis for adapting learning environments to individual learners. Such adaptation is what is sought, especially in open inquiry learning environments, as can be read in the following quote:

> "The promise offered by inquiry learning [in which students actively discover information] is tempered by the problems students typically experience when using this approach. [...] A challenge lies in adapting the learning environment to respond not only to differences between learners but also to the developing knowledge and skills of the individual learner. [...] Automating this would need an adequate cognitive diagnosis of both a student's learning process and developing knowledge and might be based on the log files of the student's interaction with the system." de Jong (2006, p. 532–533)

Inquiry learning environments encourage students to discover underlying phenomena through scientific inquiry processes like defining a hypothesis, performing an experiment, interpreting the results and drawing conclusions. Students find inquiry learning difficult. The environments offer a lot of freedom and students have to think about both the domain of learning as well as following the inquiry process: "data gathering, analysis, interpretation, and communication are all challenging tasks that are made more difficult by the need for content-area knowledge" (Edelson, Gordin, & Pea, 1999, p. 399). Unassisted inquiry learning is not effective (Mayer, 2004) and inquiry learning environments therefore provide guidance and scaffolds to increase the effectiveness of learning. A recent study (Eysink, de Jong, Berthold, Kollöffel, Opferman, & Wouters, 2009) and a meta-study (Alfieri, Brooks, Aldrich, & Tenenbaum, 2011) in which inquiry learning with additional support is compared to other educational approaches found that inquiry learning is more effective.

Given these findings, we expect that the effectiveness of inquiry learning environments can be further improved when adaptation is decided on dynamically, based on an analysis of learner activity. The general idea is to develop analytics software, called *pedagogical agents*, that continuously monitor and analyse the activity of learners with the objective to adapt the learning environment to the learner when this is appropriate. Results of the analysis can be presented to the learner in different ways: the activation of scaffolds or prompts, or by visualizing certain aspects of the learning process. In addition to tracing the activities of learners, pedagogical agents can evaluate products of the learning process (e.g., models students created), and compare these products to products of peers or normative reference objects.

Actions students perform in inquiry learning environments are stored in log files and the analysis of this data can help to understand how students use a particular inquiry learning environment and what kind of adaptation might be appropriate. Log file analysis is therefore a prerequisite for the development of pedagogical agents. More broadly, the analysis of educational data has gained considerable attention in recent years and two, closely related, research communities have emerged: *Educational Data Mining*[1] (EDM) and *Learning Analytics*[2] (LA). Educational data mining is described as "the area of scientific inquiry centered around the development of methods for making discoveries within the unique kinds of data that come from educational settings, and using those methods to better understand students and the settings which they learn in" (Baker, 2010, p. 548). Learning analytics is described as "the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimising learning and the environments in which it occurs" (Siemens, 2011, online). These definitions indicate that both disciplines aim to understand what learners do based on the traces they leave behind. In EDM this understanding is generally achieved by applying data mining techniques (clustering, classification, prediction, association rules, sequential pattern mining, text mining), and discovering relationships that describe some aspect of learning behaviour. The survey of Romero & Ventura (2007) and the Handbook of Educational Data Mining (Romero, Ventura, Pechenizkiy, & Baker, 2011) provide an overview of the application of data mining techniques to educational data. Eductional data mining starts with the educational data and tries to reveal the patterns

[1] `www.educationaldatamining.org`

[2] `www.learninganalytics.net`

it may contain. Learning analytics is broader in scope, it is "a holistic approach that combines principles of different computing areas (data and text mining, visual analytics and data visualization) with those of social sciences, pedagogy and psychology" (Ali, Hatala, Gašević, & Jovanić, 2012, p. 470). Learning analytics is more result-centered than EDM. Learning analytics starts with a motivation of what to search for and for what purposes the outcome could be used. A natural outcome of learning analytics are abstracted overviews and visualisations of the findings.

Data resulting from learning can be thought of as being "coarse" or "fine grained". An example of *coarse-grained* data is a grade for a course. This type of data is coarse grained because all the intermediate steps the learner took to obtain the grade are unknown, and can therefore not be analysed. Of course, coarse-grained learner data itself can be analysed. For example, by applying data mining techniques such as relationship or association mining, to investigate whether students who obtained high grades in one course also obtain high grades in other courses (e.g., Romero, Ventura, Espejo, & Hervas, 2008). In learning analytics, coarse-grained learner data is frequently related to other indicators that are available, for instance grades on courses related to activity in social media or prior education. *Fine-grained* data, on the other hand, consists of a "complete" trace of the activities the learner performed. Depending on the learning environment, types of actions may include selecting variable values in a simulation tool, chat messages in a collaborative environment or answers in tutoring systems.

A survey of the proceedings of the latest educational data mining (EDM 2011; Pechenizkiy, Calders, Conati, Ventura, Romero, & Stamper (2011)) and learning analytics conferences (LAK 2011; Long, Siemens, Conole, & Gašević (2011)) illustrates the differences between the two communities. All twenty full papers at EDM 2011 used data mining methods, and most papers motivated the research to improve on previous applications of data mining. Ten papers used visualisation to present the results to stakeholders (individual learners, instructors or learning institutions) and three papers mentioned "recommendation" (of learning objects or peers to cooperate with). All twenty full papers presented at EDM 2011 used fine-grained analysis, sixteen related to student modelling in tutoring systems. Of the 27 papers at LAK 2011, seven used fine-grained analysis (log files, chat analysis), nine coarse-grained analysis (data from learning management systems or learning object repositories), two used both fine- and coarse-grained analysis, and the remaining nine papers were either theoretical or provided a framework without data analysis. In conclusion, the educational data mining and the learning analytics communities are primarily concerned with data analysis and visualisation to understand learner behaviour. Some learning analytics research tries to change the behaviour of learners through visualisation of learner activities. Based on the survey, there is no evidence of active research into dynamically changing the learning environment to fit the needs of learners.

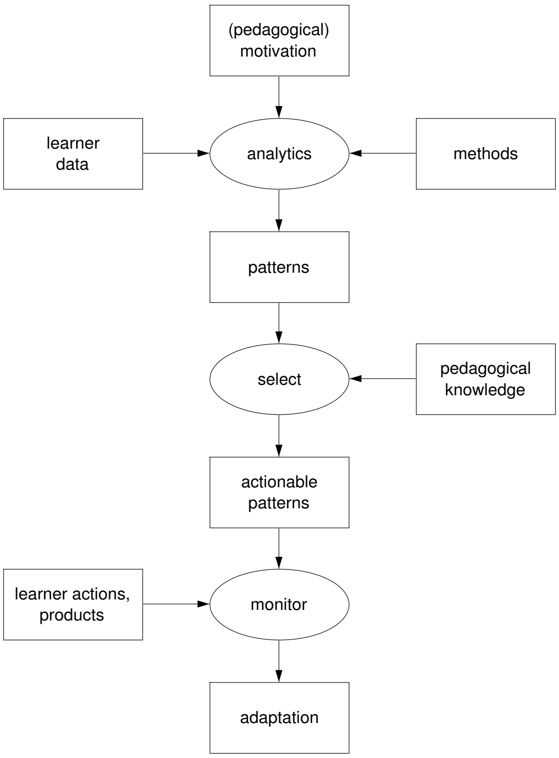


Figure 1.1: Steps to realise pedagogical agents. *Analytics* step finds patterns based on (historical) learner data. Pedagogical knowledge *selects* patterns that are actionable. Agents *monitor* learner activity and base adaptation on the actionable patterns.

In this thesis we investigate how the analytics of fine-grained data resulting from inquiry learning environments can be used to initiate adaptation of the learning environment. Figure 1.1 gives an overview of the steps required to realise this. Methods are used to find patterns in the data sets resulting from learning. To make these patterns actionable, pedagogical knowledge is applied to select relevant patterns and link these patterns to the appropriate adaptation for learners. Finally, the actionable patterns are implemented in a pedagogical agent and the agent monitors learner activity for the occurrence of the actionable patterns and initiates adaptation of the learning environment. The realisation of pedagogical agents therefore depends on several aspects: discovering patterns in data sets related to learning environments, selecting the patterns suitable to base adaptation on, and detecting the occurrence of the selected patterns.

The approach sketched in Figure 1.1 readily applies to much of current learning analytics research. A coarse-grained example is the Signals system developed and used at Purdue University (Campbell, 2007). Signals collects data from the learning management system on course materials used, sessions attended, participation in discussions, and so forth. This data is then related to students' test scores and historical data of previous students resulting in a prediction of how well a student will perform. The patterns are assessed by teachers, and are depicted as a "traffic light" (green, yellow, red) visualisation to students.

Section 1.1 presents an overview of the types of data generated and manipulated in learning environments. Learners perform actions, produce objects during the learning process and collaborate with their peers. The data that results from these activities provides a baseline to which data mining and analytics can be applied. Section 1.2 describes how the results of the analysis of learner data can be used for adaptation. Section 1.3 reviews methods which contribute to the realisation of pedagogical agents given the types of data available. These include general techniques such as data mining and text analysis, as well as log file analysis. Finally, in Section 1.4 we summarize this chapter and give an outline of the remainder of the thesis.

1.1 Data

In this section we provide an overview of the data related to learning and learning environments. Baker (2010) uses a distinction based on the context in which the data was generated and distinguishes keystroke, answer, session, student, classroom, and school level data. We make a distinction between the processes underlying the generation of the data and the types of data. This distinction is motivated by the analysis envisaged: tracking the learner interacting with the learning environment, possibly based on information about the learner (the process), and evaluating the results of learning (the products).

1.1.1 Where the data comes from

The two main sources of data related to learning are data about learners, and data that results from the interaction of a learner with a learning environment. Most learning environments store a detailed record of learner actions in log files (Hulshof, 2004) and tools inside learning environments keep track of both actions and the learning objects created as a result of the learning process. In collaborative environments, chat logs track the interactions between learners. Sometimes observational data (video, audio, eye tracking) is collected to supplement the analysis of log files (Dyke, Lund, & Girardot, 2009). Log files represent what the learner has done in the learning environment and given that they capture the "behaviour" are a primary source for analysis.

Information about the learner in general (age, gender, etc.), assessments (e.g., test scores, skills), learning histories (e.g., courses taken, classes), and presence and activeness in social media can be used to ground the process data in the log files. A common method in the learning sciences is to measure the "quality" of the behaviour in the learning environment by calculating the differences on tests before and after interaction with the learning environment. Analysis techniques can use these "quality" measures to find patterns in the log files that explain the difference in learning outcomes.

A special case is when the learning environment and assessment are intertwined. This happens when the learning environment consists of quizzes or other drill-oriented tasks (e.g., spelling a spoken word, solving a small problem, web-based courses). In these cases, the process data primarily consists of the time taken for an assignment and (the correctness of) the answer. These types of learning environments are popular in the EDM community, the process data has a relatively simple structure, comes in large volumes and can be analysed with a variety of data mining techniques (e.g., Baker, Barnes, & Beck, 2008). The Pittsburgh Science of Learning Center's DataShop[3] hosts a publicly accessible repository of such data sets (Koedinger, Cunningham, Skogsholm, & Leber, 2008).

Log files and other stored data are sufficient to support the offline analysis step in Figure 1.1. For the monitor step it is necessary that learning environments make learner actions and products available to agents in real time. More and more learning environments provide a communication infrastructure that makes this possible, see for example the blackboard software architecture described in Section 3.4.3.

[3] `http://learnlab.org/datashop`

1.1.2 What the data looks like

We distinguish four types of data related to learning that can be analysed: the learning process (activities), the objects or products learners create, communication and collaboration between learners, and data about learners (age, courses taken, etc.). The first three are the most relevant for our research and are described in further detail below.

Activities

Log files keep track of all learner-initiated interaction. They thus contribute to the "keystroke level" analysis (Baker, 2010) of educational data. Although there have been attempts to standardize the format of log files, e.g., Analog (Christoph, Anjewierden, Sandberg, & Wielinga, 2003) and Common Format (Martinez, Harrer, & Barros, 2005), the diversity of learning environments and the types of actions possible varies so widely that the lowest common denominator is to use a standardised machine-readable representation such as XML. Tatiana (Dyke et al., 2009), a tool for the analysis of computer supported collaborative learning (CSCL), also defines a proprietary format but includes filters that researchers can program to import their data.

Formally, we can view a log file as a chronologically ordered set of items where each item represents a learner action. For each item at least the following information is usually available.

Action type. The type of learner-initiated action. Common types are *answer* (to a question or an assignment), *add*, *delete*, *insert* (edit operations), *chat*, *run simulation*, *request hint*, etc. In quiz environments, the number of different action types is relatively limited (provide answer, request hint). Inquiry learning and simulation environments allow many different action types. The simulation environment SimQuest (van Joolingen & de Jong, 2003) generates more than sixty different action types, for example *start session*, *run assignment*, *change variable* and *open answer*.

Timestamp. The point in time the action occurred. Usually a precision of one millisecond is used to allow synchronization with observational data, e.g., video or EEG.

Learner. Identifier for the learner or group of learners.

Context. Most learning environments have a notion of context. The context can be related to the learning material, an assignment for example, the learning environment, a particular phase or sub-tool, or a combination of these. Context information can contribute to transition analysis, how learners navigate through the learning environment (Hulshof, 2004).

Attributes. Additional information that represents the necessary detail of an action. Obviously, there is a strong dependency on the action type. For example, a *change variable* action has the name of the variable and the new value as attributes.

The above information on learner actions potentially supports all standard types of static content analysis (frequency, coding), as well as analysis over time (sequences, transitions between contexts).

In practice, log files contain all actions the designer of the learning environment deems relevant to record. Mostow (2004) suggests to log actions at different levels of granularity to support different types of analysis. Given that it is difficult to even anticipate the types of analysis, it appears more appropriate to log at least all actions such that replay becomes possible. If the objective of the analysis is to determine learner behaviour at a more abstract level, actions not relevant to such analysis can simply be ignored. Another type of problem are actions that cause a state change in the environment. Suppose a learner wants to run an experiment with $k = 5$ and $n = 3$ (k and n are input variables). Setting these two variables may be represented as two unrelated actions in the log file, and pressing the *run experiment* button as a third. From the analysis point of view, the activity the learner wants to pursue is *run experiment*($k = 5, n = 3$). In the log file we might see *change variable*(k, 5), other actions, *change variable*(n, 3), other actions, *run experiment*, and the intended *run experiment*($k = 5, n = 3$) needs to be inferred from the action sequence.

In this section we have touched on several issues related to log files of learning environments. Technically, the representation of a log file is relevant. For computer-based analysis an XML-based representation appears most appropriate.

Products

In many learning environments students are given the task to produce something. These objects can be products of the learning process (e.g., essays, runnable models), or serve as an externalization or structuring mechanism of learner knowledge (e.g., concept maps, drawings).

Free text. Products represented as text are, for obvious reasons, very common. They can play the role of summary, report, essay, argumentation, and so forth.

Structured text. Forms or templates which the learner has to fill in are an example of structured text. Simple types are sentence openers, hypotheses and open answers, which are templates with a single field. Forms provide some guidance to learners, through the labels associated with the fields, and it may also be easier to analyse and compare learners based on forms rather than on free text.

Drawings, diagrams. Freehand drawings and diagrams can be used by learners to externalize their knowledge and help with self-explanation (Ainsworth & Iacovides, 2005). They can also be used in collaborative environments to exchange ideas with others or to obtain a common understanding (Gijlers, van Dijk, & Weinberger, 2011). Applications that support freehand drawings, for example FreeStyler (Hoppe & Gassner, 2002), are being integrated in learning environments.

Concept maps are a popular restricted type of diagrams or graph in which learners can structure their knowledge about a domain by defining relevant concepts and the relations between these concepts.

Models. Models are formal notations that are runnable. In learning environments the ability to run models is attractive because the learner immediately obtains feedback about the functioning of the model (van Joolingen, de Jong, Lazonder, Savelsbergh, & Manlove, 2005). A distinction can be made between environments in which the learner interacts with a predefined model, often referred to as simulation environments (e.g., KM Quest (Leemkuil, de Jong, de Hoog, & Christoph, 2003)), and modelling environments in which the learner has to construct a model for a given task. In simulation environments the learner's task is to understand how the underlying model works. The products are the sets of input variables the learner has manipulated. In modelling environments, the model created by the learner is the product. An example of a modelling environment is Co-Lab (van Joolingen et al., 2005) which is based on system dynamics and supports both simulation of predefined models and model construction by learners.

Data sets. A final type of product is a data set resulting from experimentation. Data sets can be an output of one tool and an input in another.

When several learners work on the same or similar tasks, object repositories result. These repositories can be used to track the progress of a single learner over time and also to compare the products of learners. In collaborative environments, the repositories can reflect progress of a group of learners and provide an opportunity for learners working together based on an analysis of their products.

Most types of products resulting from learning as listed above also occur in non-learning situations. This implies that for the analysis and evaluation methods may already exist elsewhere.

Communication and collaboration

Communication and collaboration facilities in learning environments provide a rich source of data and various opportunities for analysis. One popular method of analysis is social network analysis (SNA) (Scott, 1991) in which nodes reflect the actors (students) and edges represent the ties or social connections between actors. An example of SNA in relation to learning is "who replies to whom" on a discussion forum, the edges then represent the number of replies between students. SNA can contribute to the discovery of clusters or communities of students that share a relationship. SNA is a popular approach in learning analytics, seven of the 27 papers at LAK 2011 are about applying SNA and visualisation of social networks.

Communication and collaboration can also form the basis of content or semantic analysis. Following on from the forum example, one can try to determine whether there is a relation between the content (or topic) of forum messages and whether a given student replies. Often, collaborative environments provide a text-based chat facility and the messages students exchange can thus be analysed. Sometimes this analysis is simply counting the number of words, sometimes text analysis techniques are applied to understand what students communicate about.

Summary and discussion

In the previous sections we have presented an overview of the learner data that is available for analysis. The presentation has largely been logical: the different types of data sources and the role of these sources. The exact physical representation can be slightly different due to design choices and practical considerations. The simulation environment SimQuest, for example, represents an action type called *chat* which has the text of the message and the receiving peer as attributes. Similarly, during the development of a product, edit operations are generally sufficient to reconstruct the intermediate products. In some cases edit operations may have side effects, for example deleting a concept in a concept map causes the relations of the concept to disappear as well (without the student explicitly deleting these relations and as a consequence no *delete relation* actions).

1.2 Adaptation

This section addresses the question of how the results of the analysis of learner data (Section 1.1.2), can result in a change of the learning environment.

Adaptation based on analysed learner data can influence the learning environment and the learner in several ways. This can be implicit (changing the difficulty of the learning environment), directive (specific instructions to students), or informative (showing interaction patterns).

Implicit feedback. Based on the analysis of the data sources, the learning environment can be adapted without letting the student know. If a student makes many errors, the assignments can be made easier, or when students are working with a simulation environment, the number of variables can be reduced. Of course, the learning environment can also be made more challenging for students.

Directive feedback. Directive feedback is when students are given specific instructions. An example, based on product analysis, is suggesting missing elements in a model, or the suggestion to collaborate with a specific peer. In some intelligent tutoring systems, the student can ask for directive help by pressing a hint button.

Informative feedback. Analysis can also be used to provide students with a perspective on their own learning process. Informative feedback is usually visualised in a "dashboard". The indicators in the dashboard change dynamically depending on learner activity. Students are themselves responsible for changing their behaviour.

In the next two sections we describe interventions that occur during implicit or directive feedback and, visualisation as the primary method to provide informative feedback to students.

1.2.1 Interventions

In the learning sciences the term "scaffolding" is commonly used to refer to the adaptation of learning environments to match the skill and knowledge level of an individual learner. Scaffolds are add-ons in the learning environment that are not strictly necessary, but can help learners to focus on the learning process, e.g., a hypothesis scratchpad (Gijlers & de Jong, 2009). As mentioned earlier, some form of scaffolding is nearly always required in inquiry learning environments (Alfieri et al., 2011). Examples of how scaffolds are presented to the learner are prompts, a simplified user interface, or sequencing the order in which assignments or questions are presented. In current practice, scaffolds are permanent during a session with a learning environment. The challenge is to fade in and fade out scaffolds on the basis of activity patterns detected.

Interventions can be based on log data but also on the evaluation of products, sometimes in combination with activity analysis, particularly how long the student has been active. As mentioned in Section 1.1.2 products can range from short texts, such as open answers or hypotheses, to complex models. Evaluation can take place at the level of the structure of a product or analysing the semantics represented by the product compared to the domain of learning. An example of structural analysis is determining whether a hypothesis object contains terms that indicate it is a hypothesis (e.g., a conditional statement involving “if”, “then”). If it does not contain “hypothesis like words” then the student could be prompted to think of another hypothesis, or the learner could receive a scaffold that contains a typical syntactical pattern, “if ... then ...” to complete.

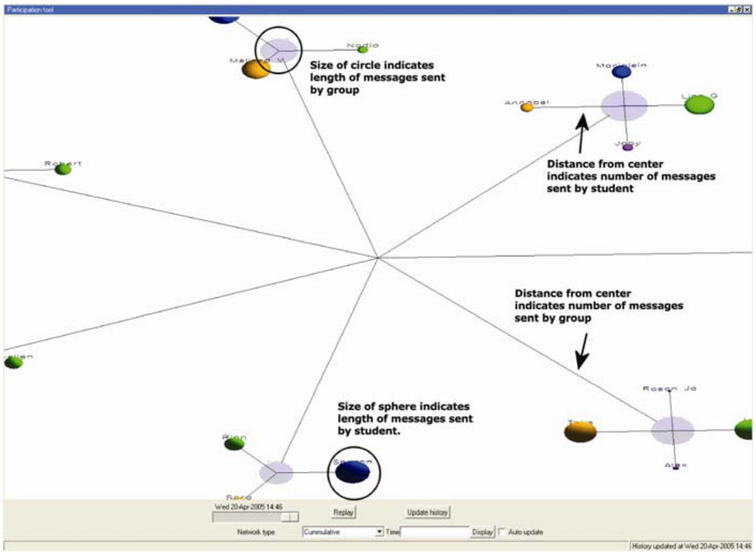
1.2.2 Information visualisation

There are many “consumers” for visualisations resulting from learner data: the individual learner, (small) groups of learners, teachers, researchers, and even learning institutions. For individual learners an important purpose of visualisation is as an awareness indicator, for instance by visualising a state or how much progress is being made.

Visualisations for groups of learners contribute to awareness with respect to relations in the group. These indicators are often visualisations of some aspect related to the learning process. For example, Janssen (2008) has identified several problems in collaborative environments: lack of awareness (of other group members), communication problems (mainly caused by using a computer to communicate), coordination problems (focusing, engagement, agreeing, etc.) and lack of quality in the discussions. He proposes to use visualisations to partly address this, for example by a participation tool (see Figure 1.2) which aims to “affect participation through motivational and feedback processes” (Janssen, 2008, p. 37–38).

Learning analytics almost exclusively relies on visualisation to communicate information to learners. “For learners [..] it can be extremely useful to have a visual overview of their activities and how they relate to those of their peers or other actors in the learning experience” (Duval, 2011, p. 12). The role of visualisations in learning environments is to help learners better understand what they are doing. These visualisations can contain either a representation of the activity of learners, for instance the number of chats, or an interpretation of the results of the activity of a learner, for instance about the content of the chats.

A simple example of such an indicator, inspired by smileys, is an avatar representation of two learners in a collaborative environment (Anjewierden, Kollöffel, & Hulshof, 2007). The shape of the two learner avatars, see Figure 1.3, changes when they exchange messages in a simulation environment. Automatic chat analysis is applied



Size of circle indicates length of messages sent by group

Distance from center indicates number of messages sent by student

Distance from center indicates number of messages sent by group

Size of sphere indicates length of messages sent by student.

Figure 1.2: Participation tool showing a visualisation of the level of communication in a collaborative environment (Janssen, 2008, p. 45, Figure 2.2).

to classify the messages as one of domain (head), regulative (body), social (arms) and technical (legs). If a domain message is typed the head becomes larger. Pedagogically, the idea is that the learners reflect on the shape of the avatar, if the head is very small and the body is very large, the suggestion is to discuss the domain of learning more.

Visualisations that represent indicators of learner activity are called dashboards. Figure 1.4 contains an example in which traditional information graphics (Harris, 1999), such as bar charts and line graphs is used. One of the most appealing visualisations of log file data is the Wattle Tree (Kay, Masionneuve, Yacef, & Reimann,

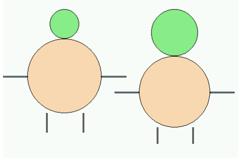


Figure 1.3: Avatars representing content of chat communication between two learners. See text for details.

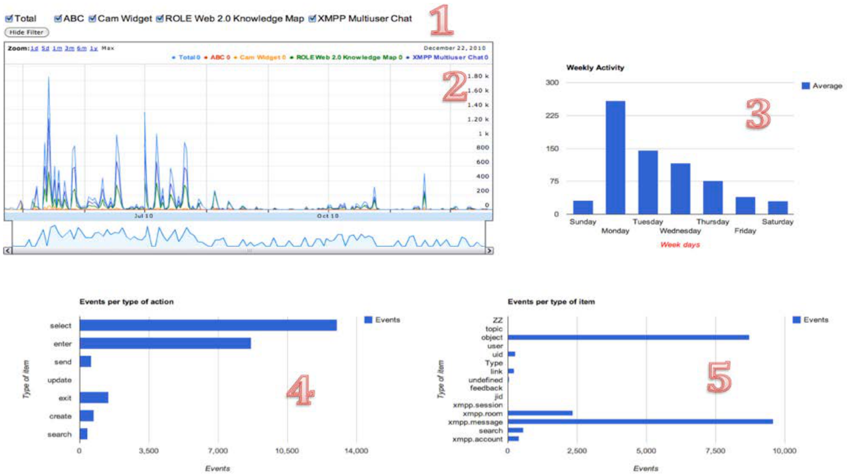


Figure 1.4: Example of a dashboard (Duval, 2011, p. 13, Figure 5).

2006a), see Figure 1.5. Here the activities of learners in a group are displayed in a time line, one for each learner running from bottom to top. Activities of the learners are visualised as yellow and orange "flowers" and green leafs. Larger flowers represent more activity, larger leafs represent that it took the learner longer to respond to a request from another learner. Kay, Masionneuve, Yacef, & Reimann (2006b, p. 7) note "It appears that group members would gain far more from all the displays than the lecturer can. In particular, each individual would have a real understanding of what their own Wattle Tree meant." This type of visualisation can be used for reflection by learners and as an overview for teachers.

In conclusion, visualisation is a powerful technique to present information about the learning process to all stakeholders involved. This is especially true when the visualisation changes dynamically. Dashboards and the indicator of participation (Janssen, 2008) are examples of dynamic visualisations that can help learners monitor their own activity. Both static and dynamic visualisations can aid researchers and teachers to understand learner behaviour.

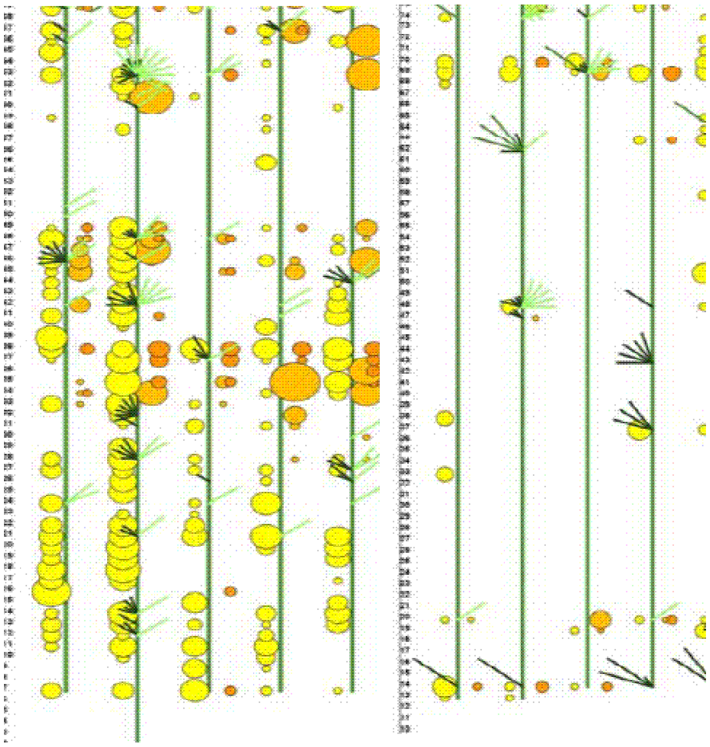


Figure 1.5: Wattle trees to represent group activity, good group (left; Kay et al. (2006b, p. 201, Figure 4)) and dysfunctional group (right).

1.3 Methods

In the following sections we describe methods to analyse learner data and, where appropriate, the relation between the methods and pedagogical knowledge.

1.3.1 Mining methods

Data mining is concerned with the automatic discovery of patterns in a set of data. Data mining is applied to large, often homogeneous, data sets to find patterns that are well-supported (frequent). Brief descriptions of commonly used data mining methods are given below.

Classification. Classification refers to the assignment of parts of the data set to predefined categories. Most classifiers are "supervised", i.e., they use an example data set from which they "learn" the parameters that determine the classification. Classification is similar to categorical coding (see Section 1.3.4).

Clustering. Clustering separates the data into subsets based on the similarity of features found in the data.

Relationship mining, association rule mining. This method is applied when an "item" has several features and associations between the features are expected. The traditional example for an "item" is the shopping basket, and the associations that can then be discovered are of the form "if A buys X and Y he is to buy Z with probability P". In education, an item can be replaced by a student, X and Y by student activity (following courses, reading learning material), and Z with succeeding on a course.

(Predictive) modelling. The objective of predictive modelling on learner data is to define or select a model that best predicts the next step or action of a student. Predictive student modelling is the dominant method in the analysis of intelligent tutoring systems.

In the first issue of the *Journal of Educational Data Mining* Baker & Yacef (2009) give an overview of the discipline, partly based on an earlier review of EDM literature published in the period 1995–2005 by Romero & Ventura (2007). Both publications provide statistics on which data mining methods are used and changes in the trends of the usage of these methods. The major shift Baker & Yacef (2009) identify is that relationship mining has declined from 43% in the early days of EDM to less than 10% based on the proceedings of the annual EDM conferences (2008–2009). Methods from psychometrics, especially model discovery, have gained in prominence from 0% in the early days to 28% recently. The explanation provided is that this increase "is likely a reflection of the integration of researchers from the psychometrics and student modelling communities into the EDM community" (Baker & Yacef, 2009, p. 8). The emphasis of EDM has seemingly shifted from "pure" data mining approaches, such as relationship mining, to student modelling approaches. The popularity of student modelling can be partly explained by the extensive use of cognitive tutors, especially in the United States. This results in large, homogeneous, publicly available data sets that can readily be analysed by modelling techniques.[4] Inquiry learning environments generate both heterogeneous data and are used on a smaller scale. This makes it more difficult to apply data mining techniques and get meaningful outcomes.

[4] See for example the KDD Cup 2010: `http://www.kdd.org/kdd2010/kddcup.shtml`

Although the relative prominence of relationship mining (e.g., using association rules) has declined in the EDM community, it is still one of the most important traditional data mining techniques used on educational data. For example, the Signals system (Campbell, 2007) depends on patterns from relationship mining.

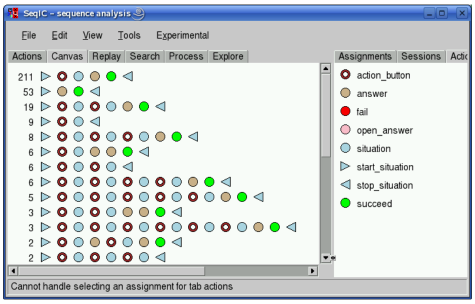
1.3.2 Frequency analysis

A standard method in the learning sciences to understand data from learning is to apply *frequency analysis*. Count the number of actions of a particular type in the data and use this count as an indicator for a certain type of behaviour. For example, in a simulation environment the number of simulations tried can be seen as a proxy for a learner's experimental or theoretical approach to learning. In a collaborative environment, the number of chats can be seen as representative for the intensity of communication with the group.

It is widely acknowledged (e.g., Rosé, Cui, Arguello, Weinberger, & Stegmann (2008); Erkens & Janssen (2008)) that frequency analysis should be used with care, as it often ignores too much relevant detail about the behaviour of a learner. For simulations it is interesting to know which values for the input variables the learner has tried. In inquiry learning trying extreme values is a good tactic, and knowing whether the learner has tried such values can be valuable input for pedagogical interventions. In a collaborative environment, the number of chats says little about the quality of the contribution, but can be used to obtain data about who talks with whom.

1.3.3 Sequence analysis

Sequence analysis takes the order in which learner actions are performed into account. Finding frequent sub-sequences and relating these sub-sequences to other information about learners is an established approach. To find interesting sequences a representation is needed that is expressive enough to capture relevant details of the learning process, but not too complex as it would reduce the likelihood of finding frequent patterns (Kay, Masionneuve, Yacef, & Zaïane, 2006). Such representations will dependent on the particular learning environment and which patterns are of interest. For example, Perera, Kay, Koprinska, Yacef, & Zaïane (2009) have used the notation (iRj) to capture a sequence in which j learners used resource R a total of i times in succession. In this abstraction the particular learners who used the resource and the order in which they did this is ignored, for example $(3R2)$ could be AAB, BBA, ABA, BAB (where A and B are learners). The abstraction (iRj) is an example of a pedagogically motivated selection of the underlying patterns from the raw data.



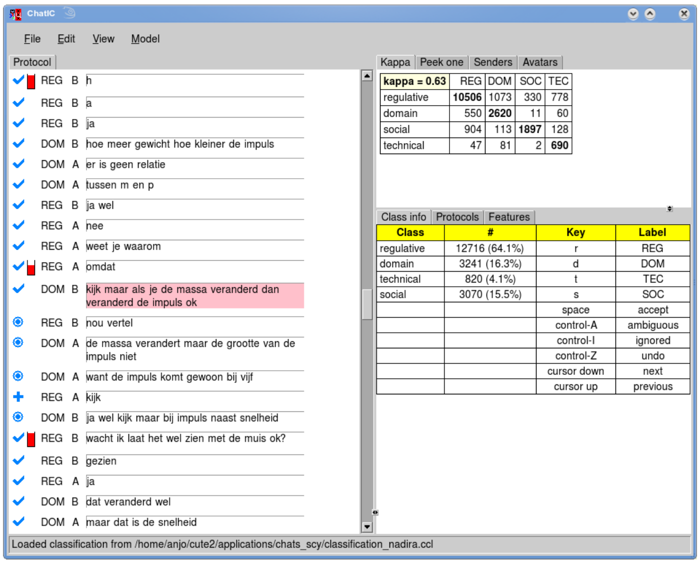
Cannot handle selecting an assignment for tab actions

Figure 1.6: Most common action sequences for an assignment. 211 learners pressed the action button to start a simulation, followed by the correct answer.

Another application of sequence analysis is to look at the order in which the learner performs actions. An illustration is provided in Figure 1.6. Right is a legend for the action types and at left sequences found in the log files from Kollöffel (2008). The most frequent sequence (211 learners) was to run a simulation (type: "action button") and next give the correct answer (type: *answer* followed by *succeed*). 53 learners did not run a simulation, but immediately gave the correct answer. This form of sequence analysis has several applications. One can develop a mathematical model of the probability that one action is followed by some other action (e.g., using Markov chains). These models can be used to predict the behaviour of future learners. A second type of application is to derive different kinds of strategies from the sequences, for example to determine whether learners run a simulation even when they know the answer.

1.3.4 Coding, abstracting and representing expert knowledge

A general method, often used in the behavioural sciences, is to assign a categorical code to learner actions. Categorical coding is necessary when the "raw data" is too heterogeneous for analysis. Applied to log files, the code is an interpretation of the action that is more abstract than the details of the action itself, but less abstract than the type of action alone. For example, all edit actions in a concept mapping tool could be coded as either improving or worsening the map. Coding schemes define the codes that are possible and give guidelines when a given code should be assigned by a human coder. A wide variety of coding schemes have been defined and applied (de Wever, Schellens, Valcke, & van Keer, 2006). After coding, frequency or sequence analysis can be applied on the codes rather than on the underlying data.



✓ 🟥	REG	B	h
✓	REG	B	a
✓	REG	B	ja
✓	DOM	B	hoe meer gewicht hoe kleiner de impuls
✓	DOM	A	er is geen relatie
✓	DOM	A	tussen m en p
✓	REG	B	ja wel
✓	REG	A	nee
✓	REG	A	weet je waarom
✓ 🟥	REG	A	omdat
✓	DOM	B	kijk maar als je de massa veranderd dan veranderd de impuls ok
◉	REG	B	nou vertel
◉	DOM	A	de massa verandert maar de grootte van de impuls niet
◉	DOM	A	want de impuls komt gewoon bij vijf
+	REG	A	kijk
◉	DOM	B	ja wel kijk maar bij impuls naast snelheid
✓ 🟥	REG	B	wacht ik laat het wel zien met de muis ok?
✓	REG	B	gezien
✓	REG	A	ja
✓	DOM	B	dat veranderd wel
✓	DOM	A	maar dat is de snelheid

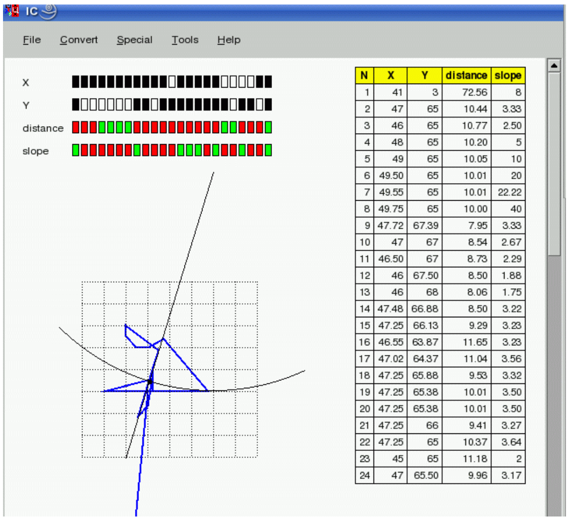
kappa = 0.63	REG	DOM	SOC	TEC
regulative	**10506**	1073	330	778
domain	550	**2620**	11	60
social	904	113	**1897**	128
technical	47	81	2	**690**

Class	#	Key	Label
regulative	12716 (64.1%)	r	REG
domain	3241 (16.3%)	d	DOM
technical	820 (4.1%)	t	TEC
social	3070 (15.5%)	s	SOC
		space	accept
		control-A	ambiguous
		control-I	ignored
		control-Z	undo
		cursor down	next
		cursor up	previous

Loaded classification from /home/anjo/cute2/applications/chats_scy/classification_nadira.ccl

Figure 1.7: ChatIC, an interactive tool researchers can use to train an algorithm to code chats from a collaborative learning environment.

For the analysis of complex structures, such as concept maps, the usual method is to define expert solutions. An agent can then compare the expert solution to the student solution. For the classification of (short) text messages, machine learning techniques are used to train the agent. Several tools that support this training and categorical coding have been developed. The most prominent is TagHelper (Rosé et al., 2008) which uses a suite of machine learning algorithms and can learn to classify chats and other short texts after human training. An alternative to TagHelper is ChatIC (Anjewierden & Gijlers, 2008). ChatIC, see Figure 1.7, has an intuitive interactive interface for training. At left are the messages prefixed with the coding (REG, ...). Each time the expert enters a code for a chat message, ChatIC updates the underlying model and then recomputes the coding for all chats. If there is a discrepancy between the code applied by the human and the algorithm a "red" indicator appears before a message. At lower left is the coding scheme, and at upper right ChatIC displays a confusion matrix of the agreement between the coder and the algorithm, including kappa. Practical results from TagHelper and ChatIC suggest that automatic chat coding, after training, results in acceptable accuracy. The avatar of Figure 1.3 is updated by automatic analysis of messages trained with ChatIC.



N	X	Y	distance	slope
1	41	3	72.56	8
2	47	65	10.44	3.33
3	46	65	10.77	2.50
4	48	65	10.20	5
5	49	65	10.05	10
6	49.50	65	10.01	20
7	49.55	65	10.01	22.22
8	49.75	65	10.00	40
9	47.72	67.39	7.95	3.33
10	47	67	8.54	2.67
11	46.50	67	8.73	2.29
12	46	67.50	8.50	1.88
13	46	68	8.06	1.75
14	47.48	66.88	8.50	3.22
15	47.25	66.13	9.29	3.23
16	46.55	63.87	11.65	3.23
17	47.02	64.37	11.04	3.56
18	47.25	65.88	9.53	3.32
19	47.25	65.38	10.01	3.50
20	47.25	65.38	10.01	3.50
21	47.25	66	9.41	3.27
22	47.25	65	10.37	3.64
23	45	65	11.18	2
24	47	65.50	9.96	3.17

Figure 1.8: Example of abstractions and replay based on raw log data (Hendrikse, 2008). See text for a description.

An example of abstraction is detecting VOTAT (vary one thing at a time). Only changing one variable and observing the effect on other variables can aid understanding the relation between the variables. Automatic data mining methods are unlikely to find patterns that correspond to VOTAT. It is therefore necessary to abstract from the raw data and recode the learner actions before VOTAT can be detected. In an assignment from Hendrikse (2008), learners had to find the x- and y-coordinates of a point at a certain distance from a start point while the line between these two points had to have a given slope. After each try, learners obtained feedback whether the distance and/or slope was correct. Figure 1.8 shows a tool to analyse the raw data from this assignment. The black arc, at lower left, represents the correct distance and the black line the correct slope, the point learners had to find is where the arc and line cross each other. Right are the (x, y) values tried by a learner, and the computed distance (correct is 10.0) and slope (correct is 3.26). The blue lines represent the "path" of the values tried, this path can be animated. At upper left are two abstractions of the data. The black and white abstraction is like VOTAT. If the x or y is unchanged compared to the previous try, the rectangle is white, otherwise it is black. Whether the distance and slope were correct is represented by the red (wrong) and green (correct) abstraction. The researcher has used this abstracted visualisation of the raw log data to obtain an understanding of the strategies learners use, which can in turn be used for adaptation.

1.4 Discussion and outline

Using the analysis of what learners do as a source for improving the learning experience has been a goal for a long time. However, examples in which the results of data mining are used in learning environments are difficult to find. Hübscher & Puntambekar write:

> "Educational data are mined with the goal to discover knowledge about the learners, educational software and other classroom interventions. Thus, the designers need to be explicit about how that knowledge is being used to redesign educational software. Yet, many of us working in the general area of educational technology too often talk about software or more general interventions at the implementation level. Staying at that level leaves the use of the data mining knowledge and its integration with pedagogical knowledge implicit." Hübscher & Puntambekar (2008, p. 97)

In other words, they suggest that EDM research should be less concerned about data mining technology and focus on addressing how the outcomes of data mining can be integrated into learning environments such that learners might benefit. Based on a survey of the full papers at EDM 2011 (Pechenizkiy et al., 2011) given at the start of this chapter there is still a focus on technology.

Learning analytics may be a more promising path. We repeat its definition: "the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimising learning and the environments in which it occurs" (Siemens, 2011, online). Learning analytics has a purpose "understanding and **optimising** learning and the environments in which it occurs". The analysis of educational data can result in an understanding of what learners are doing (which is what EDM predominantly aims at), changing the behaviour of the learner (through for example visualisation and other forms of informative feedback as exemplified by learning analytics) and dynamic adaptation of the learning environment.

In this thesis we will contribute to the latter of these challenges: the dynamic adaptation of inquiry learning environments on the basis of learner behaviour. Previous sections have described related work supportive of our research. In Section 1.2 approaches to achieve adaptation are given. These approaches have a sound foundation in the learning sciences and related disciplines, and they can be used. A major challenge is to discover actionable patterns (Figure 1.1) in the kinds of data

(Section 1.1) with the available methods (Section 1.3). Past research indicates that actionable patterns do not simply emerge from the data by applying data mining techniques. The search for actionable patterns has to be more focussed and use techniques beyond data mining. The approach we follow in the thesis is to look at each of the three types of data available (process, products, communication and collaboration) and select methods specific to find actionable patterns in these types of data.

Process. The way in which students use a learning environment can traditionally be found in log files. In modern environments, see for example de Jong et al. (2010), user actions can be made available immediately for analysis. The analysis of the actions students perform can result in patterns of unsystematic behaviour (for instance violating the VOTAT rule during experimentation), not using all resources available or discovering a student is stuck. Feedback based on process analysis is often in the form of hints or through dashboard like indicators. Chapter 2 addresses process analysis and describes techniques to find pedagogically interesting sequential patterns. These patterns can be linked to feedback to students.

Products. These are the complex structures students create in learning environments. Examples of products are models, concept maps or other graph-like structures, essays, and experimental designs. The analysis of these products is often both domain specific and dependent on the type of product. Patterns normally result from a comparison between the student product and an "expert solution" or solutions that contain misconceptions students might have. Feedback can be in the form of indicators, the number of correct concepts in a concept map for example, or by pointing out specific errors. Chapter 3 describes agent-based support for the analysis of graph-like structures occurring in instructional contexts: concept maps and (system dynamics) models.

Communication and collaboration. In many modern learning environments communication and collaboration plays an important role. Students work together by exchanging short (chat) messages, by commenting on each other's work, or creating a collaborative product. The analysis of the content of messages can be used to understand what learners discuss. Chapter 4 looks at the content of chat communication between dyads of learners who collaboratively solve assignments in an inquiry learning environment.

Chapter 5 contains a summary and discussion.