Learning in design: From Characterizing Dimensions to Working Systems

Yoram Reich
Department of Solid Mechanics, Materials, and Structures,
Faculty of Engineering, Tel Aviv University, Ramat Aviv 69978, Israel
email: yoram@eng.tau.ac.il, URL: http://or.eng.tau.ac.il:7777/
Tel: +972-3-640-7385, Fax: +972-3-640-7617

Abstract: The application of machine learning (ML) to solve practical problems is complex. Only recently, due to the increased promise of ML in solving real problems and the experienced difficulty of their use, has this issue started to attract attention. This difficulty arises from the complexity of learning problems and the large variety of available techniques. In order to understand this complexity and begin to overcome it, it is important to construct a characterization of learning situations. Building on previous work that dealt with the practical use of ML, a set of dimensions is developed, contrasted with another recent proposal, and illustrated with a project on the development of a decision-support system for marine propeller design. The general research opportunities that emerge from the development of the dimensions are discussed. Leading toward working systems, a simple model is presented for setting priorities in research and in selecting learning tasks within large projects. Central to the development of the concepts discussed in this paper is their use in future projects and the recording of their successes, limitations, and failures.

1 Introduction

The ability of humans to learn is fundamental to development and survival. The fascination with this ability has attracted many researchers in an attempt to exploit learning when building artifacts such as computer programs. Initially, these efforts concentrated in specific large expert systems projects (e.g., Teiresias for MYCIN (Davis, 1979), RULEGEN for Meta-Dendral (Buchanan and Mitchell, 1978), and later, LEAP for VEXED (Mitchell et al, 1985)) and their objective was very practical — to solve the knowledge acquisition bottleneck of building these systems. Over time and in a bottom up fashion, two major research thrusts have emerged: (1) creating computer programs for solving some task by learning from data instead of by programming, and (2) creating programs that improve their performance speed. The mutual goal of the two thrusts was improved performance: solving problems better and faster. As time passed, the emphasis of these thrusts shifted from the practical aspects of solving problems to the theoretical issues underlying learning mechanisms; these were often studied on simplified problems.

ML in design research begun in the opposite direction. Simple problems were formulated and researchers tried to solve them with available machine learning (ML) techniques (Arciszewski et al, 1987; Lu and Chen, 1987; McLaughlin and Gero, 1987). One study presented a bottom-up analysis of six such projects carried out at the EDRC, CMU at the time (Reich et al, 1991). The six projects differed in their underlying assumptions about the scope of learning: from automated to human directed. The goal of the analysis was to abstract some fundamental roles of ML in design from the particular project experiences while attending to practical needs. The analysis identified human learning as a key to progress in engineering and design and listed four core learning processes (see Figure 1):

(1) *Machine learning.* The availability of a large knowledge base and an experience base supports using a variety of machine learning techniques for converting design experiences into design knowledge to be used by computational tools or humans.
(2) *Knowledge Acquisition.* The availability of expert designers who are willing to share their knowledge provides an opportunity to use a variety of knowledge acquisition and other tools to transfer knowledge from the experts into knowledge bases. The knowledge once encoded and managed is available for use by computational tools and further dissemination.
(3) *Human learning (including formal teaching).* Facilities for browsing, visualizing, exploring, and analyzing large repositories of design experiences and knowledge bases support the ability of humans to comprehend their content and assimilate valuable design knowledge. In addition, facilities that aid interaction with the design system, e.g., devices that support exploration of the design space via experimentation, support human learning of the particular problem.
(4) *Organizational shared memory creation.* Facilities that organize and manage design experiences, design knowledge, and other data from multiple designers and systems into coherent structures can support the creation of organizational or disciplinary knowledge and aid in its dissemination (Konda et al, 1992).

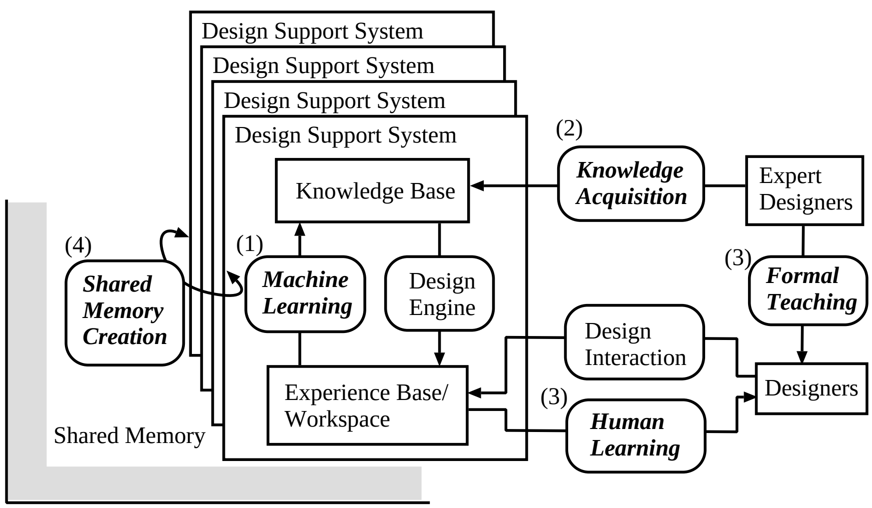

Figure 1: A system perspective of learning in design

Figure 1 shows the relationships between the learning activities in a design setting composed of designers and design support systems. In addition to the shown arrows, designers have full control over the workings of all the learning and design mechanisms.

This study continues and stresses the focus on the practical objective of ML: "The aim of computational research with respect to learning should ... be to produce computational environments that fit and enhance human practice." (Reich et al, 1993, p. 165). After reviewing the roles that ML could play in design (section 2), Section 3 refines a previously developed process of using ML

(Reich et al, 1993; Reich, 1997) by identifying eight dimensions, henceforth denoted by *RD*, that characterize ML tools for enhancing human design practice. Section 3 also contrasts these dimensions with a set of dimensions recently proposed by Grecu and Brown (1996) that will be denoted by *GBD*. This set is one amongst several others that appeared in the past (e.g., Persidis and Duffy, 1991; Reich et al, 1991; Waldron, 1991). Section 4 illustrates the use of the dimensions in one project and Section 5 concludes the paper.

2 Solving practical learning problems with ML

The four learning processes discussed in the introduction define the possible interaction schemes between a learning design system components. Each of these processes can be realized in a variety of ways. This section analyzes two general roles assigned to ML and the steps and issues related to applying ML tools to solve learning problems.

2.1 Functional roles of ML

In order to set the ground for understanding the roles of ML in design it is instrumental to analyze the functionality of ML programs and the roles that ML programs can assume abstracted from any particular task domain. The common task of ML has always been the generation of knowledge (or models) from examples (in the form of decision rules, e.g., CN2 (Clark and Niblett, 1989) or trees, e.g., C4.5 (Quinlan, 1992)) for predicting some missing values of a new example. Examples could be the inspection data of bridges and the prediction would be whether a newly inspected bridge needs maintenance work within the next six months. This role of ML is the *performance* role.

Figure 2(a) depicts the process of building models using ML techniques and subsequently using them to make predictions in the performance role. When used for prediction, the best model possible should be constructed and validated using appropriate statistical tests. In order to create good performance models, large data is required. Performance models can be comprehensible, thus allowing some debugging or verification to take place, but can also be incomprehensible thus operating as black boxes.

Figure 2(b) presents the process of building models, and using them to better understand the data used as input or its source. The process is the same as for the performance role except that instead of obtaining the best prediction the user controls the learning process in order to inspect results and gain insight about the data. For example, when attempting to understand a complex decision procedure, one can generate traces of the procedure and its recommendations, and use several ML programs to learn multiple simple models of the procedure (Reich et al, 1996b). This process is sometimes called metamodeling (Blanning, 1975). The models need not be good, they can even be poor, as long as they are comprehensible and lead to better understanding of the data. Since quality of models from a statistical perspective is not required, this role can be employed even with small data. This role of ML is the *understanding* role recently, it has branched off from ML to form its own discipline: data mining or knowledge discovery from databases. Table 1 summarizes the two ML roles.

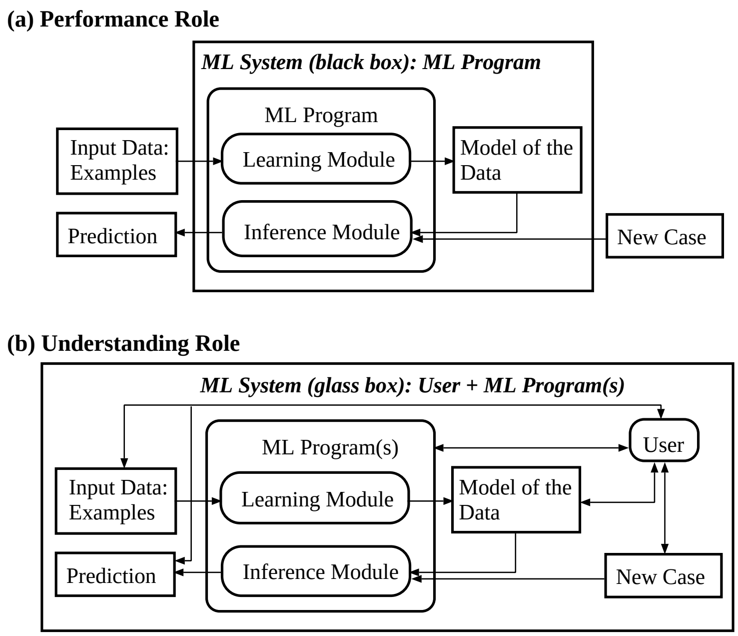

Figure 2: Performance and understanding roles of ML

Table 1: Performance versus understanding roles of learning

Learning role	Performance	Understanding
Who is learning	Programs	Humans
Input	Large data	Data
How is learning performed	Best ML program	Different ML programs
Knowledge representation	Symbolic + "sub-symbolic"	Symbolic
Nature of process	Structured, algorithmic	Ill-structured

2.2 Process of using ML techniques

Only recently has the study of practical use of ML been recognized as critical by the general ML community (Kodratoff et al, 1994; Brodley and Smyth, 1996). The two ML roles are two extreme specific instances of using ML. A more refined analysis revealed a seven-step process called *Contextualized ML Modeling (CMLM)* which is shown in Figure 3 (Reich, 1997).

There is significant variety built into the seven-step process. Each step can be executed in different ways. Thus, following Ashby's law of requisite variety (Ashby, 1958), this process has the potential to match the significant variety in learning contexts. The learning context manifests in step 1. The remaining steps deal with how this context is addressed. In order to better understand the nature of learning contexts a set of dimensions is developed.

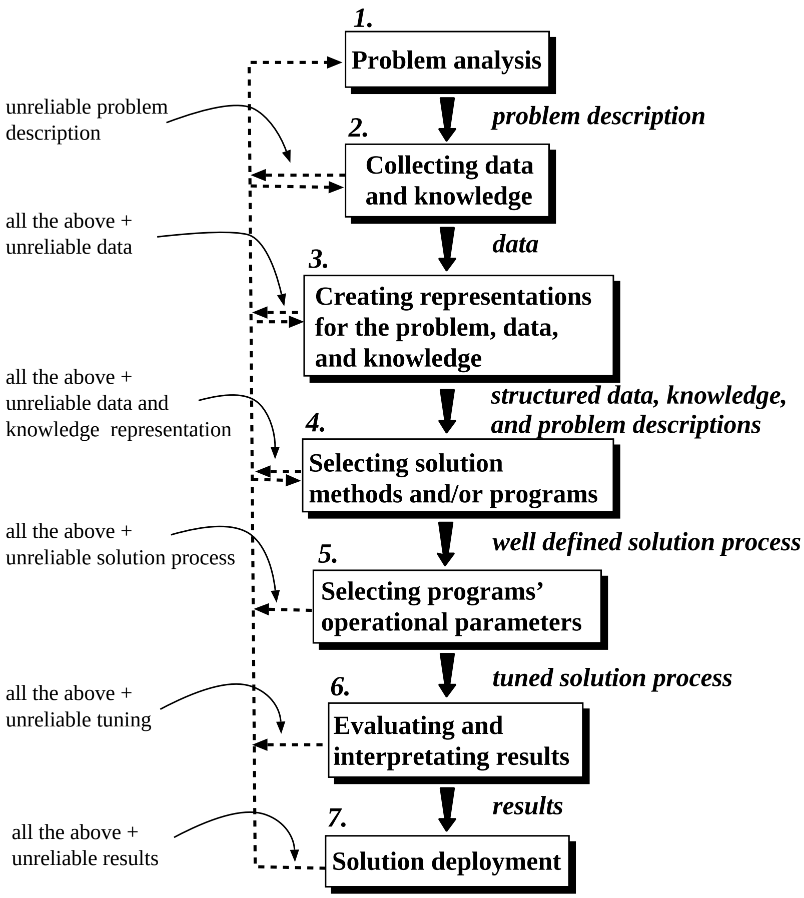

Figure 3: CMLM: A model of ML use in practice (Reich, 1997)

3 Dimensions of learning contexts: Top down analysis of learning in design

Learning contexts deal with learning tasks or activities as they are embedded in a particular engineering practice and organization. In general, any task or activity can be associated with an actor, the reasons for action, the prerequisite for action, the way the action takes place, its results, and the resources required to carry it out. The application of these items to a learning task leads to the following questions:

(1) *Learners:* Who is learning?
(2) *Finiteness of knowledge:* Why does the learner want to learn?
(3) *Timing:* When does the learner learn and when are the results of learning needed?
(4) *Activities:* What is the learner doing?
(5) *Improvements:* What is learned?
(6) *Prerequisites and Processes:* How does the learner learn?
(7) *Outcome:* What are the consequences of learning?
(8) *Cost:* How much resources are needed to carry out the learning activity?

Table 2: Summary of ML in design dimensions

Reich's dimensions (RD)		CMLM steps		Grecu & Brown's dimensions (GBD)	
1	Who is learning	1	Problem analysis	6	Local vs. global learning
2	Why does the learner want to learn	1	Problem analysis	1	Trigger for learning
3	When does the learner learn and when are results needed	1	Problem analysis	—	—
4	What is the learner doing	1	Problem analysis	—	—
5	What is learned	1	Problem analysis	3	What might be learned
6	How does the learner learn	2-5	Collecting data, creating representations, selecting programs and parameters	2, 4, 5	Elements supporting learning, availability of knowledge, and methods
7	What are the consequences of learning	6, 7	Testing and system deployment	7	Consequences of learning
8	How much resources are needed to carry out the learning activity	—	—	—	—

The answers to these questions can serve as dimensions for describing the space of learning situations. While the 7-step process concentrates on the development of solutions to learning problems, these dimensions focus on the nature of the problems. The relation between the steps and dimensions is shown in Table 2. From the table, new dimensions (RD) are more elaborate than Grecu and Brown's dimensions (GBD). The next subsections, each discusses one dimension of RD. Where applicable, after each paragraph that raises issues related to the dimensions there follows a paragraph that discusses the *supporting ML tools or techniques* for the issues raised. Each subsection concludes with a brief discussion on the relation of the dimension and GBD.

3.1 Learners: Who is learning (in the context of design)?

An agent is referred to as a *primary* learner if it can extract knowledge from data and use it while solving real problems. According to the four core learning mechanisms reviewed in the introduction there can be two primary learners. The *machine learning* process corresponds to the case where *ML programs* are the primary learners. ML programs have been the focal learners in design research. The *knowledge acquisition* and *human learning* processes assume *designers* as the primary learners. Less attention was given to these processes (exceptions include (Duffy and Duffy, 1996; Dong and Agogino, 1997; Reich et al, 1991; Reich et al, 1993; Waldron, 1991)). Note that in participatory design, designers are whomever participate in the project including customers (Reich et al, 1996a).

The *organizational shared memory creation* process focuses on the learning of *a group of designers*. This process results in disciplinary knowledge in forms such as design codes and manuals, mathematical models, and organizational memory that captures organizational culture, practices, or language. The complex and critical organizational aspects that often determine the fate of projects are central to this process yet, almost no work has been done in ML in design on this process.

When ML are the primary learners, the role of learning is performance. When a designer or designers learn, the role of ML is understanding. Designers as primary learners may employ ML as mechanisms of learning which would be considered the secondary learners.

Relationship to GBD: This dimension relates to item 6 in GBD: the "Local vs. Global Learning"; however, it does not distinguishes between one or several ML programs if they use exactly the same vocabulary or even if they do not — in real design, a major part of the problem is learning about other participants' perception of the problem, forming interaction schemes, and common vocabularies. Knowledge of these aspects is as part of design knowledge as are other types of knowledge such as mathematical models. Thus, learning to interact is not separable from learning domain knowledge.

3.2 Finiteness of knowledge: Why does the learner learn?

There are several fundamental reasons that force whoever is engaged in technological activity to learn. They are derived from the dynamic nature of the environment and our limited resources to engage in learning.

(1) *Ontological reason*: The dynamic nature of the environment forces people to continually adapt. In design contexts, people's needs change, technology change, and with them, the nature of design problems and design knowledge change.
Supporting ML tools/techniques. Learning incrementally and detecting change in an environment is important to knowing when is it time to learn. ML work on this subject includes adapting to changing domains (Reich and Fenves, 1992; Schlimmer and Granger, 1986) and the potential use of natural language processing (NLP) to detect terminological changes in a domain (Reich et al, 1993; Dong and Agogino, 1997).

(2) *Epistemological reason*: One cannot know all knowledge. Knowing means engaging in some interaction with the environment - an activity that requires resources. Limited resources prevent one from knowing all knowledge even in one discipline. Thus, in order to solve problems even in a quasi-static environment, one is constantly learning additional knowledge relevant to the present design.
Supporting ML tools/techniques. Knowledge acquisition (KA) and ML tools (in the understanding role) are capable of assimilating new knowledge. However, ML could be further used to learn the limits of particular knowledge and to predict the resources for learning knowledge in a domain with particular characteristics. These can support planning to address future learning needs.
Knowledge is also subjective. Even a consensus between design participants is only appreciated subjectively by group members (Reich et al, 1996a). Theories are also subjective because our understanding of their working and applicability is subjective (Subrahmanian et al, 1993). The problem of multiple model or multiple language reconciliation is critical for learning and communication. Two agents will never use the same language or meaning of terms unless they were carefully crafted by the same person and unless a very strict vocabulary use has been enforced — a task practical only for simple problems. A proof of the complexity of this

task are the difficulties encountered in the CYC project over the years.
Supporting ML tools/techniques. NLP can help agents construct common shared languages. Communication also plays a critical role as demonstrated by Steels (1997). He showed how robots with communication channels, start without a common language, and gradually built terms and mutual understanding of objects through interaction.

(3) *Behavioral reason* (which is a part of the epistemological reasons): One can only perform as fast as one can. Improved performance on tasks (including learning how to learn better) allows for spending less resources on previous tasks and spending additional resources on solving new problems and on learning.
Supporting ML tools/techniques. Explanation-based learning (EBL) (Mitchell et al, 1986) is a mechanism for narrowing search including failure avoidance. EBL can be costly (Minton, 1988), thus, will benefit from learning to predict the cost of learning.

The ontological, epistemological, and behavioral reasons emphasize the central role of learning in problem solving. In particular settings, *explicit* learning is exercised when it becomes apparent that something is deficient (e.g., failure of a product). In real design, a failure may take a long time to feed back into practice or even to be detected (e.g., Quebec Bridge) and its fix may be costly (e.g., failure of the Tacoma Narrows Bridge and the subsequent strengthening of bridges such as the Golden Gate Bridge). It is often hard to locate the sources of failures; they could be beyond the scope of theoretical knowledge, domain practice, or personal expertise. It is sometimes even unclear how to determine that a failure occurred and how to act upon it.

Supporting ML tools/techniques. When using a DSS, all activities performed with the system could be recorded and ML could analyze them. The availability of historical records presents opportunities to understand the sources of failures and the limitations of available design knowledge and procedures. In addition, historical records could be analyzed to help the adaptation of the programs to their users needs and thus, improve their usability.

Relationship to GBD: This question relates to the "trigger of learning" in GBD . Instead of listing various specific triggers, the present dimension starts with the fundamental reasons underlying learning and focuses on the central role of failure as a driver for design advances (Petroski, 1989).

3.3 Timing: When does the learner learn and when are the results of learning needed?

This dimension is influenced by the previous dimension. People mostly learn pro-actively by formal training in schools and universities. Later on, they learn when a need arises, if they have spare time, or by consciously allocating resources. For example, people may learn while planning a new project when they understand that it involves new technologies or the use of new design tools.

People learn unconsciously while performing design or explicitly when introspecting on last activities or on the whole process. If recent activities led to a failure or shortcomings of the design, then the learning experience has been expensive. It is therefore, cost effective to detect when certain decisions are at the limit of existing expertise and learn in advance. Although pro-active learning is most cost effective, both timings are critical for improved problem solving capabilities.

Supporting ML tools/techniques. In order to determine when to allocate resources for learning, one could learn the applicability and reliability of knowledge. This may allow for planning pro-active learning activities instead of leaving them to the last minute. It is also useful to learn the cost of learning in order to determine when is it best to learn.

Relationship to GBD: There is no corresponding dimension in GBD. However, ML mechanisms described in item 5 of RD and other ML techniques could be classified into traditional ML types: (1) pro-active learners are those using concept learning (supervised and unsupervised) and (2) reactive learners are those using explanation-based learning, knowledge compilation, case-based, analogical learning, or other techniques. Nevertheless, this classification is a simplification because a primary learner can learn reactively by employing a learning system that is a batch (or a pro-active) learner.

3.4 Activities: What is the learner doing?

Learning is interwoven in design problem solving. The design context determines the learning context (e.g., representation language, bias, background knowledge, and consequently, the complexity of the learning task). Thus, it is important to know what is the activity that an agent performs when it is learning. Maintaining the context for future reference is a critical and very hard undertaking (Reich et al, 1993).

Design activities could be classified into traditional problem-solving methods. They can be useful if analyzed at the right granularity. To illustrate this issue consider finite-element analysis. Although it is analysis at the highest level, it involves synthesis tasks amongst others: the generation of mesh and the selection of appropriate element types. More generally, if by performing a task analysis, large tasks are decomposed into smaller tasks, one can better grasp the large tasks by integrating the solutions to the smaller tasks (Reich, 1991).

Supporting ML tools/techniques. In ML, very syntactic characterization of data has been developed in order to associate it with the success of classification algorithms to learn from such data. The association was done using learning as well (Gama and Brazdil, 1995; Michie et al, 1994). Thus, meta-learning was used to create rules for predicting which algorithm can be best for which domain characterization. In design domains, the characterization is more complex but the benefits could be significant if there would be a mapping between such a characterization and appropriate ML tools. This was the goal of CMLM.

Relationship to GBD: There is no corresponding dimension in GDB.

3.5 Improvements: What is learned?

This item is closely related to the discussion in sections 3.2, 3.3, and 3.4. Insight about the role of ML in design originates from observing the kinds of things that are learned in real design processes.

(1) First, designers learn *technical (in most cases analytical) knowledge.*
Supporting ML tools/techniques. KA and ML techniques are obvious methods for learning technical knowledge. The discovery of analytical knowledge from data presents new opportunities to speed up learning. Such discovery would be more appealing if ML tools could explain the origin of the knowledge they generate.

(2) Second, in the course of designing, designers *learn about the particular problem* they solve. They can develop a language for discussing the problem, and can uncover implicit constraints, interactions, or trade-offs, between the problem parameters or objectives. The knowledge uncovered during design is contextualized in particular situations, it may be generalizable or not.

Supporting ML tools/techniques. Many techniques can assist in this learning activity starting with KA, ML, or NLP tools, continuing with tools for knowledge organization, and finishing with tools for data modeling, visualization, and analysis.

(3) Third, designers *assimilate experiences* for use in future design problems. These experiences include whatever designers can remember about the design processes. Such experiences and especially their organization differentiate experts from novices (Chi et al, 1988). Designers must always be aware that new design situations prevent the "as-is" application of previous experiences. Designers need to learn *the similarities* as well as *the differences* between current and previous problems and adapt old solutions to new situations.

Supporting ML tools/techniques. Tools for information or knowledge organization can be more effective to users if they can learn to tune to their users. One example involves improving the ability of case-based reasoning (CBR) to index cases based on their performance and users requests.

(4) Fourth, from user feedback or from the failures of artifacts, designers learn about the viability of certain design beliefs, judgments, decisions, or practices in certain situations. Feedback from users (customers) about design arrives after a product is released to the market, thus, tends to be hard to use.

Relationship to GBD: This dimension relates to item 3 in GBD: "What might be learned?" However, item 3 focuses on aspects expressed in AI concepts that can be learned by programs. Without specifying these aspects in detail, and independent of the way design is modeled, candidates for learning are *all aspects of these models.*

3.6 Prerequisites and processes: How does the learner learn?

Humans as primary learners learn in a variety of ways as discussed before. However, in order to be effective learners, they require certain conditions. For example, they can benefit from certain *felicity conditions* (VanLehn, 1983) that make sure that newly learned knowledge extends existing knowledge by a small margin that could be integrated well with what the learner already knows. The same conditions can be used when designing specific ML programs or for training existing incremental learners.

Human learning also benefits from mechanisms that organize information and knowledge. It is easier to associate new information with organized body of knowledge. In order to analyze information and comprehend it, search facilities as well as different modeling and visualization facilities can be used. Humans can also benefit from collaborative learning environments (Kaye, 1991).

Supporting ML tools/techniques. When ML programs are the primary learners or if they are used as secondary learners, the process of using them to solve learning problems follows CMLM (Section 2.2). Most applications of ML have used supervised and unsupervised concept learning techniques and not the more complex techniques (Langley and Simon, 1995). Even so, there still is a significant variety of ML techniques, each with a variety of parameters and alternative ways of learning. Consequently, it is important to gradually meta-learn a refined characterization of ML programs and a mapping between different learning tasks and ML programs.

Relationship to GBD: This dimension combines the second, fourth, and fifth dimensions of GBD and also deals with human learning. Also, the fourth dimension corresponds to the 2nd step in CMLM, the fifth dimension to the 4th step, and the second dimension to parts of steps 1, 2, 6. Specific details on these steps can be found in (Reich, 1997).

3.7 Outcome: The consequences of learning

This is one of the most critical dimensions. The ultimate goal of learning is the improved performance of design systems including designers and programs. A practical learning context will determine the required improvements such as increased design speed or improved design quality. Research projects tend to be less precise or stringent in setting their goals and often, the consequences of learning are established following simple demonstrations.

One in-depth discussion on testing focused on statistical testing of models created by ML (Reich, 1997). Another important way of testing involves the solution of benchmark problems which have to be constructed carefully to be useful. They need to be used carefully since their results do not necessarily generalize. Another testing involves the creation of artificial design problems that model real design problems. Since their characteristics vary at will, they provide ways to obtain detailed understanding of the working of ML programs. If complicated systems are developed in research they can hardly be replicated. Therefore, it is important that not only test problems are shared but also ML programs. For example, Ecobweb (Reich and Fenves, 1992) was made available and already used by different research groups.

Testing the results of learning in a real setting is more complex. Ultimately, any learning problem is solved only once. Therefore, the mode of research and development follows a case-study approach (Reich, 1994) and the testing of results is multidimensional (Reich, 1995).

Relationship to GBD: This dimension corresponds to item 7 in GBD, however, the present dimension stresses practical consequences and the development of valid measures.

3.8 Cost: How much resources are needed to carry out the learning activity?

One dimension that is often neglected is the cost of learning. It is highly dependent on the previous dimensions. Figure 4 shows a simple conceptual picture of fictitious ML projects. Each ML project can be ranked based on its ease of development or solvability and the importance of the design problem it solves. The more solvable is a ML problem, the less expensive it is to implement. The more solvable is a ML problem, the more complex and important issue could be tackled given a fixed development cost.

As mentioned before, it is beneficial to use ML to predict the cost of learning. With this estimated cost, and given a constraint on a project budget, and the anticipated benefit from using ML, one could maximize the total benefit by a simple procedure after recognizing that this maximization is an instance of the knapsack problem in operations research (Winston, 1994). In order to solve

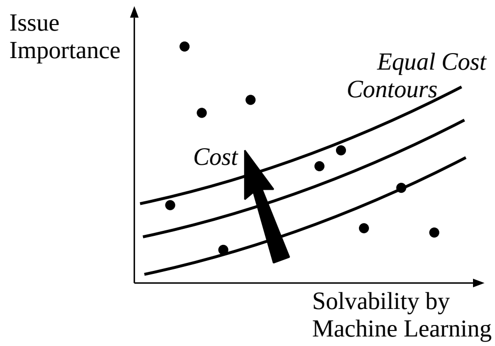

Figure 4: Cost-effectiveness of ML projects

this problem, the potential ML solutions need to be sorted based on a benefit/cost index and they are allocated resources according to this index until the budget is exhausted. This ordering relies on a simple model but it is useful for preliminary analysis. Further, the same model can shed light on setting priorities at several levels:

(1) Priorities in selecting critical research topics.
(2) Priorities in solving ML problems in a particular design context.
(3) Priorities in applying specific learning strategies in multistrategy programs.

Relationship to GBD: There is no corresponding dimension in GDB.

3.9 Research issues

Each of the dimensions raises issues that were matched by supporting ML tools or techniques that promise to be valuable research topics. The explication and organization of these issues is an example of the usefulness of creating the dimensions. The issues summary is shown in Figure 5. Each row denotes a function that ML needs to perform and each column includes an aspect such as a mechanism, facility, or data, that can support these functions. The issues in the figure are meant to be generally useful to support ML in design and not specific tasks such as approximating a function or constructing a classification. The supporting aspects, in particular, the history recording, clearly suggest the use of an integrated approach to research and development of ML in design because they are best realized within a comprehensive design system (Figure 1).

Most studies on ML in design have attempted to tackle specific topics or dimensions of ML in design by solving point problems. However, general understanding of ML in design cannot emerge from addressing issues in isolation, or as Newell said (1973) "you cannot play 20 questions with nature and win". The reason being that decomposition does not eliminate complexity, it merely moves it to the boundaries therefore, general understanding must emerge from the integration of the parts.

In contrast, an integrative approach to research risks spending too much effort on the research infrastructure and less on the particular research problem. A combination of the two approaches might be most beneficial: For quick examination of issues, simple problems and isolated solutions could be developed in a divide-and-conquer strategy, but to gain real insight, these solutions must

ML Functionality \ Supporting Facility, Method, or Data	Record history	Records of learning situations	NLP	Incremental/adaptable learning	Domain characterization	Meta-learning (about learning and users)	Communication	Explanation facility
Detect change in a domain	+		+	+	+			
Detect limits/applicability of knowledge/method	+			+	+	+		
Detect failure in design/learning	+	+				+		
Predict resources of learning	+	+		+		+		
Improve subjective understanding	+	+	+				+	+
Reconcile language	+	+	+				+	
Be usable	+				+	+		+

Figure 5: Matching ML functionality with supporting methods

be integrated.

4 Example

A brief description of one application, currently under development, illustrates: (1) the eight dimensions; (2) the complexity of, and benefits from, classifying a real design problem that involves ML with a set of pre-defined characteristics; and (3) the setting of development priorities. The application involves developing a DSS for propeller design (Reich et al, 1997) which is a knowledge intensive reasonably well structured task. The system is intended to be deployed in an organization that amongst other activities, designs marine propellers.

Who is learning? The answer is not straight forward. First, the people involved in the development of the DSS are learners. The development of such a system is a design activity and as such, all parties involve learn about propeller design, technology for building DSS, and methods for developing and deploying DSS. Learning takes place in the interaction between the different participants and in the interaction between researchers' theories and the practical needs of users. The system is developed to allow propeller designers to have quick access to maximum information and knowledge relevant to their design. In addition, it is developed to allow other design participants to carry out several design tasks themselves without interacting with propeller designers. Therefore, propeller designers and other participants are expected to learn through using the DSS. For very specific learning tasks such as function approximations, ML would be the primary learners. Finally, given that several designers from the target organization will use the system, the organization will also learn throughout the project. The introduction of new computational technology into this organization will have to be done incrementally, making sure that each tool demonstrates its practical utility.

Why does the learner want to learn? The purpose of building the application is to improve the propeller design process of the organization by incorporating expertise accumulated over many years of experience into a DSS. This corresponds to the epistemological reason applied at the organization level. The critical aspect is shortening the design cycle thus saving money and becoming more competitive. This points to a behavioral reason, again at the organization level. Since the domain is stable, the ontological reason for learning is secondary. A DSS approach that supports the recording and maintenance of design history provides a basis for continuous improvement of the design process and knowledge. At a more specific level, some of the propeller design tasks are based on data analysis of old data that could be improved continually using ML. Other tasks are based on using previously designed and manufactured propellers thus could benefit from techniques such as CBR. Presently, the developers are learning in order to better understand the nature of data and knowledge available for building the system.

When does the learner learn and when are the results of learning needed? Most learning activities in this project can be planned in advance pro-actively. First, the developers learn before building the DSS. Second, specific simple design tasks will be automated by using ML programs to create the required knowledge. The designers could learn pro-actively before starting a particular design by inspecting information about previous similar designs. Subsequently, during design, they would learn reactively by observing the intermediate results of the system and through interacting with it. Occasionally, some learning mechanisms will be executed off-line to update the knowledge. At the organization level, learning throughout the project will take place when prototypes of the DSS are deployed. Thus learning will be reactive and will have to be planned for carefully.

What is the learner doing? The three sets of primary learners — developers, designers, and ML programs — are engaged in building the DSS, design propeller, and perform specific design subtasks, respectively. Concentrating on propeller designers, their task is to design a propeller including its complex geometry given customer requirements about ship performance, and subject to physical and regulatory constraints. In order to better understand this task it is decomposed into seven subtasks as performed in the target organization (Reich et al, 1997):

(1) Prepare for model experiments by performing efficiency calculations and selecting the most suitable stock propeller for model testing.
(2) Estimate the effective wake distribution by applying several possible scaling methods to the model test results.
(3) Determine propeller profile thickness according to classification society rules.
(4) Perform lifting-line and lifting-surface calculations to determine a detailed description of propeller geometry.
(5) Smooth the geometry obtained in step 4.
(6) Perform final hydrodynamic analysis involving all physical phenomena and perform local redesign if necessary.
(7) Check the strength of the propeller against classification society rules and perform local redesign if necessary.

The above decomposition is also too general and can be further detailed into subtasks that could be supported with computational tools from AI, ML, or other disciplines. Four task types emerged:

(1) Mapping between some numeric input data to numeric output related by some non-linear mapping (e.g., estimating parameters, performing numerical analyzes).
(2) Selecting between available alternatives (e.g., material properties, modeling code).
(3) Performing local modifications with small overall impact (e.g., geometry smoothing).
(4) Verifying the design against design codes, constraints, etc.

These tasks could be further elaborated with specifications of the knowledge used in each instantiation of the task.

What is being learned? The developers of the DSS learn about the process of designing propellers. The development project helps designers reflect upon their own process and often improve upon it. This process will further be refined in response to the introduction of the DSS and throughout its life-cycle. The designers will learn a variety of issues while they design and ML programs will learn knowledge to assist in the four general tasks mentioned in the previous paragraph.

How does the learner learn? The developers learn about the problem by communicating with designers and by testing various tools on sample databases and subproblems. Prototyping is fundamental to this learning. Several techniques that could be used to implement the learning tasks were identified. For example, history capture will help designers learn by providing complete traceability of the process. They will also learn by easy access to previous designs and through the use of CBR. Neural networks or instance-based learning will be used for various function mappings.

What are the consequences of learning? The consequence of process is the anticipated creation of a knowledge-base integrated with other knowledge and numerical analyzes codes that will increase the speed and reliability of design. The resulting DSS is expected to capture and collect vital organizational knowledge. In order to ensure project success new tools will be introduced if they are well established and even then, only incrementally.

Cost: How much resources are needed to carry out the learning activity? The seven design tasks or the four task types identified in the problem analysis step present opportunities for developing a variety of support mechanisms based on ML, AI, or other disciplines. One method — CBR integrated with rule-base reasoning — could be used to implement most of the functionality required (Reich et al, 1997). This will allow to minimize the cost of the software infrastructure.

A preliminary ordering of the design tasks to address can be based on a qualitative estimation of their benefit/cost index, where cost included the effort to develop an application and the potential acceptability by the organization. Four tasks have been identified and ordered according to this index:

(1) Selecting a stock propeller for model testing.
Benefit: Significant. The selection quality has significant impact on the usefulness of the model tests. Also, once supported, the task could be assigned to the project manager in the ship model department thus shortening unnecessary communication paths.
Cost/difficulty: Moderate. This effort includes solving the problem of function approximation which is presently explored.
(2) Provide a general solution to support propeller redesign.
Benefit: Significant. Approximately half the design time is spent on redesign.
Cost/difficulty: Not trivial. CBR might provide a framework to support some redesign tasks, but the generalizability of the solution to all propeller redesign tasks is unclear.

(3) Support the smoothing process.
Benefit: Moderate. Smoothing is a rather tedious labor intensive task although not too knowledge intensive.
Cost/difficulty: Moderate. Initial analysis suggests that heuristics for smoothing can be extracted. The implementation of this task will also rely on CBR.

(4) Select methods for scaling model test results.
Benefit: Moderate. This is not a time consuming task but its appropriate execution provides valuable input for subsequent design steps. Improved quality of the results is the goal.
Cost/difficulty: Initial analysis suggests that this is a harder task than the previous.

5 Conclusions

In any scientific discipline the process of classification is fundamental and important. Building on previous research on the and practical use of ML, a top-down analysis of the role of learning in design was performed. This analysis yielded a characterization of learning situations composed of eight dimensions. In the process of creating the dimensions interesting general opportunities for ML were formulated. They were organized as functions that ML should support with some associated facilities, methods, or data that could support them. These opportunities are best realized within an integrative approach to supporting ML in design. As part of the dimensions, a method for prioritizing the work on these issues as well as selecting which ML tasks to solve in a given project was presented.

In order to illustrate the dimensions and the related issues, they were used to describe an existing project. Clearly, it is not trivial to categorize real problems with a fixed set of dimensions. Nevertheless, such analysis if done carefully and at the right level of detail, allows to better select tools to support the tasks that solve the problem.

In order to improve our understanding of the scope of ML in design, the dimensions, the general research issues, and their prioritization, need to be further developed and used when conducting research and participating in real projects. Reports of successes, as well as failures, in using these ideas will provide feedback and improve them.

6 Acknowledgments

This research was partially supported by The Israeli Ministry of Science and the Arts. The Marine Propeller DSS system project is performed in collaboration with Volker Bertram, Jürgen Friesch, and S. V. Barai.

References

Arciszewski, T., Mustafa, M., and Ziarko, W. (1987). "A methodology of design knowledge acquisition for use in learning expert systems." *International Journal of Man-Machine Studies*, 27(1):23–32.

Ashby, R. (1958). "Requisite variety and its implications for the control of complex systems." *Cybernetica*, 1(2):1–17.

Blanning, R. W. (1975). "The construction and implementation of metamodels." *Simulations*, 24:177–184.

Brodley, C. E. and Smyth, P. (1996). "Applying classification algorithms in practice." *Statistics and Computing*, (in press).

Buchanan, B. G. and Mitchell, T. M. (1978). "Model-directed learning of production rules." In Waterman, D. A. and Hayes-Roth, F., editors, *Pattern-Directed Inference Systems*, New York, NY, Academic Press.

Chi, M. T. H., Glaser, R., and Farr, M. J., editors (1988). *The Nature of Expertise*, Lawrence Erlbaum Associates, Hillsdale, N.J.

Clark, T. and Niblett, P. (1989). "The CN2 induction algorithm." *Machine Learning*, 3(4):261–283.

Davis, R. (1979). "Interactive transfer of expertise: acquisition of new inference rules." *Artificial Intelligence*, 12(2):121–157.

Dong, A. and Agogino, A. M. (1997). "Text analysis for constructing design representations." *Artificial Intelligence in Engineering*, 11(2):65–75.

Duffy, S. M. and Duffy, A. H. B. (1996). "Sharing the learning activity using intelligent CAD." *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, 10(2):83–100.

Gama, J. and Brazdil, J. (1995). "Characterization of classification algorithms." In Pinto-Ferreira, C. and Mamede, N., editors, *Progress in Artificial Intelligence, 7th Portuguese Conference on Artificial Intelligence, EPIA-95*, pages 189–200, Berlin, Springer Verlag.

Grecu, D. L. and Brown, D. C. (1996). "Dimensions of learning in agent-based design." In *Preprints of AID'96 ML in Design Workshop*.

Kaye, A. R., editor (1991). *Collaborative Learning Through Computer Conferencing*, Springer-Verlag, Berlin.

Kodratoff, Y., Moustakis, V., and Graner, N. (1994). "Can machine learning solve my problems?." *Applied Artificial Intelligence*, 8(1):1–31.

Konda, S., Monarch, I., Sargent, P., and Subrahmanian, E. (1992). "Shared memory in design: A unifying theme for research and practice." *Research in Engineering Design*, 4(1):23–42.

Langley, P. and Simon, H. A. (1995). "Applications of machine learning and rule induction." *Communications of the ACM*, 38(11):55–64.

Lu, S. C.-Y. and Chen, K. (1987). "A machine learning approach to the automatic synthesis of mechanistic knowledge for engineering decision-making." *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, 1(2):109–118.

McLaughlin, S. and Gero, J. S. (1987). "Acquiring expert knowledge from characterized designs." *Artificial Intelligence in Engineering Design, Analysis, and Manufacturing*, 1(2):73–87.

Michie, D., Spiegelhalter, D., and Taylor, C. C. (1994). *Machine Learning, Neural and Statistical Classification*, Ellis Horwood Publishers, Chichester, England.

Minton, S. (1988). "Quantitative results concerning the utility of explanation-based learning." In *Proceedings of AAAI-88*, pages 564–569, St. Paul, Minnesota, Morgan Kaufmann.

Mitchell, T., Mahadevan, S., and Steinberg, L. (1985). "LEAP: A learning apprentice for VLSI design." In *Proceedings of The Ninth International Joint Conference on Artificial Intelligence, Los Angeles, CA*, pages 573–580, San Mateo, CA, Morgan Kaufmann.

Mitchell, T. M., Keller, R. M., and Kedar-Cabelli, S. T. (1986). "Explanation-based generalization: a unifying view." *Machine Learning*, 1(1):47–86.

Newell, A. (1973). "You can't play 20 questions with nature and win: Projective comments on the papers of this symposium." In Chase, W. G., editor, *Visual Information Procesing*, NewYork, NY, Academic Press.

Persidis, A. and Duffy, A. (1991). "Learning in engineering design." In Yoshikawa, H., Arbab, F., and Tomiyama, T., editors, *Intelligent CAD, III*, pages 251–272, Amsterdam, North-Holland.

Petroski, H. (1989). "Failure as a unifying theme in design." *Design Studies*, 10(4):214–218.

Quinlan, J. R. (1992). *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA.

Reich, Y. and Fenves, S. J. (1992). "Inductive learning of synthesis knowledge." *International Journal of Expert Systems: Research and Applications*, 5(4):275–297.

Reich, Y., Coyne, R., Modi, A., Steier, D., and Subrahmanian, E. (1991). "Learning in design: An EDRC (US) perspective." In Gero, J., editor, *Artificial Intelligence in Design'91, Proceedings of The First International Conference on Artificial Intelligence in Design, Edinburgh, UK*, pages 303–321, Oxford, UK, Butterworths.

Reich, Y., Konda, S., Levy, S. N., Monarch, I., and Subrahmanian, E. (1993). "New roles for machine learning in design." *Artificial Intelligence in Engineering*, 8(3):165–181.

Reich, Y., Konda, S. L., Levy, S. N., Monarch, I. A., and Subrahmanian, E. (1996). "Varieties and issues of participation and design." *Design Studies*, 17(2):165–180.

Reich, Y., Medina, M., Shieh, T.-Y., and Jacobs, T. (1996). "Modeling and debugging engineering decision procedures with machine learning." *Journal of Computing in Civil Engineering*, 10(2):157–166.

Reich, Y., Bertram, V., and Friesch, J. (1997). "The development of a decision support system for propeller design." In *Proceedings of the 9th International Conference on Computer Applications in Shipbuilding (ICCAS '97)*.

Reich, Y. (1991). "Design knowledge acquisition: Task analysis and a partial implementation." *Knowledge Acquisition*, 3(3):237–254.

Reich, Y. (1994). "Layered models of research methodologies." *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, 8(4):263–274.

Reich, Y. (1995). "Measuring the value of knowledge." *International Journal of Human-Computer Studies*, 42(1):3–30.

Reich, Y. (1997). "Machine learning techniques for civil engineering problems." *Microcomputers in Civil Engineering*, 12(4):307–322.

Schlimmer, J. C. and Granger, R. H. J. (1986). "Beyond incremental processing: tracking concept drift." In *Proceedings of AAAI-86*, pages 502–507, Philadelphia, PA, Morgan Kaufmann.

Steels, L. (1997). "Constructing and sharing perceptual distinctions." In van Someren, M. and Widmer, G., editors, *Machine Learning: ECML-97*, pages 4–13, Berlin, Springer-Verlag.

Subrahmanian, E., Konda, S. L., Levy, S. N., Reich, Y., Westerberg, A. W., and Monarch, I. A. (1993). "Equations aren't enough: Informal modeling in design." *Artificial Intelligence in Engineering Design, Analysis, and Manufacturing*, 7(4):257–274.

VanLehn, K. (1983). "Felicity conditions for human skill acquisition: Validating an ai-based theory." Technical Report CIS-21, Xerox Palo Alto Research Centers, Palo Alto, CA, November.

Waldron, M. B. (1991). "Design processes and intelligent computer-aided design (ICAD)." In Yoshikawa, H., Arbab, F., and Tomiyama, T., editors, *Intelligent CAD, III*, pages 51–75, Amsterdam, North-Holland.

Winston, W. L. (1994). *Operations Research: Application and Algorithms*, Duxbury, Belmont, CA, 3rd edition.