Available online at www.sciencedirect.com
SCIENCE DIRECT®

Engineering Applications of Artificial Intelligence 17 (2004) 393–400

Engineering Applications of
ARTIFICIAL INTELLIGENCE

www.elsevier.com/locate/engappai

Approximation space for intelligent system design patterns

James F. Peters*

Department of Electrical and Computer Engineering, University of Manitoba, 15 Gillson Street, ENGR 504, Winnipeg, Manitoba, MB R3 T 5V6, Canada

Abstract

This article introduces an approximation space for graded acceptance of proposed models for intelligent system design relative to design patterns that conform to a design standard. A fundamental problem in system design is that feature values extracted from experimental design models tend not to match exactly patterns associated with standard design models. It is not generally known how to measure the extent that a particular intelligent system design conforms to a standard design pattern. The rough set approach introduced by Zdzisław Pawlak provides a ground for concluding to what degree a particular model for an intelligent system design is a part of a set of a set of models representing a standard. The basic assumption made in this research is that every system design can be approximated relative to a standard, and it is possible to prescribe conditions for the construction of a set of acceptable design models. It is also possible to measure the degree that a proposed set of design models is a member of a set of design models that conform to a standard. The neuron and sensor behavioral design patterns are briefly considered by way of illustration of design model approximation. A satisfaction-based approximation space for patterns extracted from intelligent system design models is introduced.

Keywords: Approximation space; Design model; Design pattern; Intelligent system; Rough sets

> The rightness of the form depends … on the degree to which it fits the rest of the ensemble.
> —Christopher Alexander, (1964)

1. Introduction

This article introduces an approach to classifying models for intelligent system design in the context of a satisfaction-based approximation space defined in the context of rough sets (Pawlak, 1982, 1991). Considerable work has been done on approximation spaces in the context of rough sets (Skowron, 2001; Skowron and Stepaniuk, 1996, 2001; Stepaniuk, 1998; Peters et al., 2002) as well as generalized approximation spaces (Polkowski, 2002; Pal et al., 2004a, b), which is directly related to a paradigm for approximate reasoning called rough mereology (Polkowski and Skowron, 1994, 1996) and recent work on pattern recognition (Skowron and Swiniarski, 2004). It is well known that experimental models for system design in general and intelligent system designs in particular seldom exactly match what might be considered a standard. This is to be expected, since system designs tend to have an unbounded number of variations relative to an accepted design pattern. Consider, for example, the variations in the implementation of design patterns in architecture made possible by pattern languages (Alexander, 1964, 1979, 2002; Alexander et al., 1977). This is expected and encouraged. It is this variation in actual system designs that is a source of a difficult classification problem. This problem is acute in reverse engineering a legacy system. It is not generally known how to measure the extent that a particular system design conforms to a standard. It is usually the case that the feature values of a particular intelligent system design, for example, *approximately* rather than *exactly* match a standard pattern. An approach to a solution of the system design classification problem is proposed in this article in the context of rough sets and a satisfaction-based form of approximation space.

In general, a behavioral model for a system design is represented by a set of interacting objects where each object is an instance of a class (a description of a set of

*Corresponding author. Tel.: +1-204-474-9603; fax: +1-204-261-4639.
E-mail address: jfpeters@ee.umanitoba.ca (J.F. Peters).

0952-1976/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2004.04.012

objects that share the same attributes, operations, and semantics). A pattern is a conjunction of feature values that are associated with a decision rule. In particular, a system design pattern is a conjunction of feature values relative to the structure and functionality of a set of classes used in designing components of a system. Patterns commonly found in models for intelligent system designs can be gleaned from class, interaction, and other diagrams (Peters, 2003; Peters and Ramanna, 2003) from UML, the Unified Modeling Language (OMG, 2001), especially in the context of systems engineering (see, e.g., Holt, 2001). In this article, only collaboration diagrams are considered.

This paper has the following organization. Basic concepts from rough sets and UML underlying the proposed approach to classifying intelligent system design models are briefly presented in Section 2. Sample system design features and design patterns are briefly considered in Sections 3 and 4, respectively. An approximation space for design patterns is considered in Section 5. A framework for classification of intelligent system design models within a satisfaction-based approximation space is given in Section 6.

2. Basic concepts

This section covers some fundamental concepts in rough sets and UML that underlie the presentation in the later sections of this paper.

2.1. Rough sets

The rough set approach introduced by Pawlak (1991) provides a ground for concluding to what degree a set of design models representing a standard are a part of a set of candidate design models. In this section, we briefly consider several fundamental concepts in rough set theory, namely, set approximation and attribute reduction. For computational reasons, a syntactic representation of knowledge is provided by rough sets in the form of data tables. Informally, a data table is represented as a collection of rows each labeled with some form of input and each column is labeled with the name of an attribute that computes a value using the row input. Formally, a data (information) table IS is represented by a pair (U,A), where U is a non-empty, finite set of objects and A is a non-empty, finite set of attributes, where $a: U \to V_a$ for every $a \in A$. For each $B \subseteq A$, there is associated an equivalence relation $\mathrm{Ind}_{\mathrm{IS}}(B)$ such that $\mathrm{Ind}_{\mathrm{IS}}(\mathrm{B}) = \{(x, x') \in U^2 \mid \forall a \in B.\ a(x) = a(x')\}$.

If $(x, x') \in \mathrm{Ind}_{\mathrm{IS}}(B)$, we say that objects x and x' are indiscernible from each other relative to attributes from B. The notation $B(x)$ denotes a block of B-indiscernible objects in the partition of U containing X. For $X \subseteq U$, the set X can be approximated only from information contained in B by constructing a B-lower and B-upper approximation denoted by $B_*(X)$ and $B^*(X)$, respectively, where $B_*(X) = \{x \in U \mid B(x) \subseteq X\}$ and $B^*(X) = \{x \in U \mid B(x) \cap X \neq \emptyset\}$. A lower approximation $B_*(X)$ of a set X is a collection of objects that can be classified with full certainty as members of X using the knowledge represented by attributes in B. By contrast, an upper approximation $B^*(X)$ of a set X is a collection of objects representing both certain and possible uncertain knowledge about X. Whenever $B_*(X) = B^*(X)$, the collection of objects can be classified perfectly, and forms what is known as a crisp set. In the case $B_*(X)$ is a proper subset of $B^*(X)$, then the set X is considered rough (inexact) relative to B.

2.2. UML collaboration diagrams

In this article, models for intelligent system designs are limited to what are known as collaboration diagrams in UML (OMG, 2001). Briefly, a collaboration diagram represents either sequential or parallel interactions between a collection of instances of classes (called objects). Objects in such a diagram are represented by boxes. An object label is of the form "object name: Class name", where the object name is optional. An association between two objects is denoted by a straight line. An interaction between two objects is denoted by a label of the form "<numeral>: [condition] message →". The numeral indicates the position of a message in a sequence of interactions that define a behavior. The "[condition]" denotes an optional boolean condition that must be satisfied before an interaction between objects can occur. The arrow head points towards the object that is stimulated by a message. An object name inside guillemets ≪ ≫ denotes a stereotype. A collaboration diagram reveals the behavior of objects interacting with each other, which is helpful in visualizing how objects work together in any system design.

Example 2.2.1 (Collaboration diagrams for neuron and sensor models). Collaboration diagrams for forms of neuron and sensor models are given in Fig. 1. In this diagram, there are three interactions in a neuron model for an adaline (adaptive linear element) that is trained

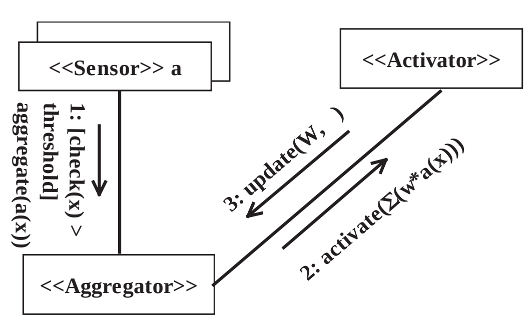

Fig. 1. Sample neuron model.