Application of Category Theory in Model-Based Diagnostic Reasoning*

Renwei Li and **Luís Moniz Pereira**
Centre for Artificial Intelligence, UNINOVA and
Department of Computer Science, U. Nova de Lisboa
2825 Monte da Caparica, Portugal
{renwei | lmp}@fct.unl.pt

Abstract

This paper discusses the use and importance of category theory in system descriptions for model-based diagnostic reasoning. We argue that category theory provides a precise and convenient conceptual language and tool to model complex systems. System descriptions are regarded as categories, in which each object is a system model and each morphism represents changes and transformations of system models. A system model is in turn viewed as a category together with some faithful functors. The category in a system model describes the structure of the device being diagnosed and the faithful functors describe the behaviour of components of the device. The morphism (transformation of system models) in the category as system description is a functor between categories describing structures of the device being diagnosed. We also discuss several special transformations of system models. From a system model a first-order theory can be obtained so that existing algorithms and software systems can then be used to generate diagnosis candidates.

1 Introduction

In model-based diagnostic reasoning systems [9], knowledge about the structure and behaviour of systems being diagnosed is provided so that normal behaviour of the systems can be predicted and compared with observed behaviour. If there is a discrepancy between predicted and observed behaviour, model-based diagnostic reasoning systems can determine which components are malfunctioning. Reiter [14] showed that there is a general diagnosing algorithm to compute diagnosis candidates, which actually serves as a logical foundation of general diagnosis systems such as GDE [3]. As Selman and Levesque showed in [15], it is NP-hard to decide if there is an explanation containing a given hypothesis. In order to generate diagnosis candidates more practically and quickly, some efforts have been reported in, e.g [4, 12, 5]. In Reiter [14] it was assumed that the description of a system being diagnosed is fixed and not changed during the whole process of diagnosis. As observed and discussed in [17, 1, 7], for several reasons we may regard diagnosis as a process, during which the system description may vary. Multiple system models seem to play a central role in diagnosis as a process. Under different diagnostic assumptions or taking different diagnostic strategies, we may have different system models [16, 17, 7]. After a certain system model is picked up, we can follow, e.g. [14, 3] to compute the diagnosis candidates.

This paper discusses the use and importance of category theory in system descriptions for model-based diagnostic reasoning. We argue that category theory provides a precise and convenient conceptual language and tool to model the process of diagnosing complex systems. A system description SD is regarded as a category, in which each object is a system model and each morphism represents change/transformation of system models. A system model SM is in turn viewed as a category St together with some faithful functors $Ok, Ab_1, \ldots, Ab_k$. The category St in a system model is the structural description which defines what the structure is and how its components are connected to each other. The faithful functors $Ok, Ab_1, \ldots, Ab_k$ describe the behaviour of the components of the device. The functor Ok is intended to capture normal behaviour of components in St, and $Ab_1, \ldots, Ab_k$ capture abnormal behaviours. Each functor Ab_i corresponds to a fault mode. The morphism (transformation of system models) in SD between system models is a functor between categories describing structures of the device being diagnosed. We will discuss several special

*Published in: John Stewman (ed.), *Proc. 8th Florida Artificial Intelligence Research Symposium*, Melbourne (FL, US), 1995, pp123-127

transformations of system models and show a normal form of regular transformations. From a system model a first-order theory can be obtained so that existing algorithms and software systems can then be used to generate diagnosis candidates. Diagnosis is viewed as a process of activities of working on SD. For each object in SD we may use existing methods to generate diagnosis candidates. For several purposes we may need to navigate in SD. The tangible achievements of this paper are as follows: (i) We provide a mathematical mechanism for better representation of the whole process, via uniform and powerful abstractions; (ii) We provide new conceptual tools that clarify and formalize previous informal descriptions of the diagnostic process. The categorical methodology proposed in this paper may be more important than other technical contributions.

Minimal knowledge about category theory, e.g. category, functor, universal construction (product object, etc.) is assumed. The reader is referred to Mac Lane [11] for category theory and its diagrammatical presentation. The rest of the paper is organized as follows. In Section 2 we use examples to show that structures of systems can be regarded as categories. In Section 3 we discuss how to categorically model systems. In Section 4 we discuss how to use existing algorithms to generate local diagnosis. In Sections 5 we show that various models of a given system can be related to each other by transformations. In particular, all system models and transformations constitute a category, called the system description. We also discuss several interesting transformations, and give a theorem on normal form of regular transformations. In Section 6 we briefly discuss diagnosis as a process.

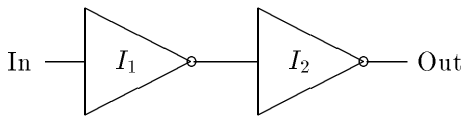
2 Structure descriptions as categories

Model-based diagnosis consists in that malfunctioning components of a system are inferred from the *structure* and *misbehaviour* of the system. In other words, the structural information of the system does play *some* role in the process of diagnosis. However, describing the structures simply as first-order sentences loses the richness of structural information about systems.

In many domains, a system consists of constituents. A constituent may have some inputs and outputs. The behaviour of a constituent may be represented by its input and output. Let's represent a constituent by an abstract arrow from the input to the output. Observe that the output of a constituent C_1 may be the input of another constituent C_2. The overall behaviour of C_1 and C_2 may be represented by the input of C_1 and the output of C_2. That is to say, the joint behaviour of C_1 and C_2 is that of some other constituent which is the *composite* of the individual arrows of C_1 and C_2. This is exactly the composition operation on morphisms of categories. This observation provides a basis for categorical representation of knowledge on structures of a system. Once the structural knowledge is represented by a category, we can employ the abstract and powerful concepts and tools of category theory in diagnosis. In particular we can use functors to relate different categories. This further provides a basis for categorical representation of changes of system models. In addition we can use faithful functors to generate first-order theory, and hence can use existing diagnostic algorithms to compute diagnoses.

In what follows we give two examples for categorical representation of structures of systems.

Example 2.1 (Two Inverters) This is a system composed of two inverters:

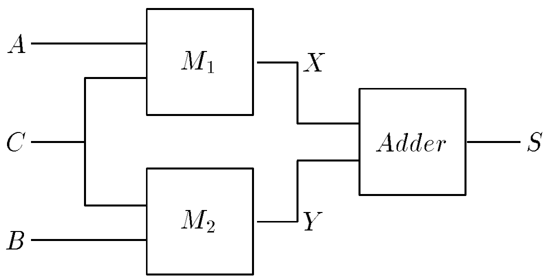


Its structure can be simply described by the following category:

$$IN \xrightarrow{I_1} X \xrightarrow{I_2} OUT$$

The identity morphisms and composite morphisms are omitted in the diagram above. Later on we will often omit all identity morphisms and composite morphisms. The morphisms I_1 and I_2 can be imagined to represent two components whose behaviour will be represented by using behavioural functors to be discussed later. The morphisms I_1 and I_2 are also called structural or external morphisms. Other morphisms such as identity morphisms are called internal morphisms. Each structural morphism corresponds to a part of the system. Properties of external morphisms need to be given by the user through behavioural functors to be discussed later, while properties of internal morphisms can be inferred from category theory itself. Later on we will not distinguish between morphisms and arrows.

Example 2.2 (MMA) This is a device consisting of two multipliers M_1 and M_2, and an adder A.



In the device above, there are three inputs A, B, C, and an output S. For this multi-input device, we can still use a category to describe its structure. Part of the arrows of the category is shown in the following diagram:

$$\begin{tikzcd}
A \times C \arrow[rr, "M_1"] & & X & \\
(A \times C) \times B \arrow[u, "\pi_1"] \arrow[d, "h"'] \arrow[rr, "N"] & & X \times Y \arrow[u, "\pi_1"'] \arrow[d, "\pi_2"'] \arrow[r, "Add"] & S \\
B \times C \arrow[rr, "M_2"'] & & Y &
\end{tikzcd}$$

where $h = (\pi_2, \pi_1 \circ \pi_2)$, $N = (\pi_1 \circ M_1, h \circ M_2)$ and π_1 and π_2 are the projection morphism.

3 System models

Note that all sets and functions constitute a category **SET** with sets as objects and functions as morphisms. To each function we can associate a first-order theory. For example, let $Boolean$ be the set $\{0, 1\}$, then the negation function $not : Boolean \to Boolean$ can be characterized in a straightforward way: $not(0) = 1 \wedge not(1) = 0$, which is the normal function of an inverter.

Definition 3.1 (System Model) A system model SM is a tuple $\langle St, Ok, Ab_1, \ldots, Ab_k \rangle$, where St is a category, and $Ok, Ab_1, \ldots, Ab_k$ are faithful functors from St to **SET**.

In $SM = \langle St, Ok, Ab_1, \ldots, Ab_k \rangle$, St is intended to describe the structure of the system, the functor Ok is intended to capture normal behaviour of components in St, and $Ab_1, \ldots, Ab_k$ capture abnormal behaviours. The functors $Ab_1, \ldots, Ab_k$ correspond to fault modes [13]. For generality we do not require the completeness of fault modes [18, 6]. All the functors $Ok, Ab_1, \ldots, Ab_k$ are generally called *behavioural functors*. If we do not want to consider fault modes, the system model will simply be $\langle St, Ok \rangle$.

Certainly, we can also make use of the category **REL** with all small sets as objects and binary relations as arrows. All functors in system models are regarded as faithful functors from St to the category **REL**. Thus our categorical model is actually in line with Struss' relational model.

Example 3.2 (MMA) A system model SM for MMA of natural numbers can be defined to be $\langle St, Ok \rangle$, where St is the category shown in Section 2, and Ok is the following functor:

$$Ok(w) = \aleph, \text{ for } x \in \{A, B, C, X, Y, S\}$$
$$Ok(M_1) = Ok(M_2) = \times, \quad Ok(Add) = +$$

where $+$ and $\times$ are the usual arithmetic operations on natural numbers $\aleph$. In the example above we omitted the compositional arrow.

The example above should be sufficient to show how to add fault modes to components.

4 Generation of diagnoses

The main and central task of diagnosis is to tell which components are malfunctioning when the observed behaviour disagrees with the normal and expected behaviour. A diagnosis is usually defined to be a set of components which are malfunctioning. In order to make representatives of components explicit, we assume a system model SM is expressed as something like $\langle St, Ok, Ab_1, \ldots, Ab_k; COMP \rangle$, where $COMP$ is a subset of morphisms of St corresponding to components.

In this section we want to show that we can still make use of existing algorithms and software systems for diagnosis. We do this by indicating that from a system model $SM = \langle St, Ok, Ab_1, \ldots, Ab_k; COMP \rangle$ we can generate a first-order theory which is the same as those in the literature such as [14]. Then everything is straightforward.

Consider the two inverters without fault modes. The system model SM for two inverters is $\langle St, Ok; COMP \rangle$, where $COMP = \{I_1, I_2\}$. Note that $Ok(I_1) = Ok(I_2) = not$, and not is a function axiomatized as follows: $not(0) = 1 \wedge not(1) = 0$. Then transform $not(0) = 1 \wedge not(1) = 0$ into

$$\neg AB(i_n) \to (not(0) = 1 \wedge not(1) = 0) \qquad \text{for } n = 1, 2$$

where $AB(i_n)$ is introduced to correspond to $Ok(i_n)$. The sentence above is just what we want. Taking it as the system description in the sense of [14], we can generate diagnosis candidates if we are given observations. The details may be found in [14, 8]. In practice people might like to use the endpoints of components instead of functions. In this case, we can simply transform it into equivalent formula:

$$\forall x, y : \neg AB(i_n) \to (y = not(x) \to$$
$$((x = 0 \to y = 1) \wedge (x = 1 \to y = 0)))$$

If we always take x as the input and y as the output, i.e., y = not(x) always holds, then we can simplify the above formula as follows:

$$\neg AB(I_n) \rightarrow ((x = 0 \rightarrow y = 1) \wedge (x = 1 \rightarrow y = 0))$$

The idea above also applies to more complex examples. We will not go into deeper discussions.

5 System descriptions

Given a system we may have different system models under different assumptions or from different perspectives. In this section we discuss relationships between two such models. We will formally discuss the concept of *transformations* of system models.

Definition 5.1 (Transformation) Given two system models $SM = \langle St, Ok, Ab_1, \ldots, Ab_k \rangle$ and $SM' = \langle St', Ok', Ab'_1, \ldots, Ab'_m \rangle$, a system model transformation Ω from SM to SM' is defined to be a functor from St to St'.

Proposition 5.2 All system models for a given device and system model transformations constitute a category, denoted by SD and called system description for that device.

The proof of the above proposition is omitted. It signifies that we can make use of categorical constructions to study diagnosis. In the following we give some special system model transformations. For the space limitation we will not delve into more profound and complicated discussions.

In all the following definitions we assume that two system models $SM = \langle St, Ok \rangle$ and $SM' = \langle St', Ok' \rangle$ are given.

Definition 5.3 (Renaming) A transformation Ω from SM to SM' is said to be a renaming if it is an isomorphic functor $\Omega : St \rightarrow St'$ such that (i) for each object S in St, $Ok(S) = Ok'(\Omega(S))$; (ii) for each arrow f in St, $Ok(f) = Ok'(\Omega(f))$.

Definition 5.4 (General Arrow) Given a system model $SM = \langle St, Ok \rangle$, the arrow d of St is said to be the general arrow of St iff if there is another arrow $k \circ d$ or $d \circ k$ then k is an identity arrow.

Definition 5.5 (Structural Refinement) A transformation Ω from SM to SM' is said to be a structural refiner if (i) it is an injection functor from St to St'; (ii) for each object $a \in St$, $Ok(a) = Ok'(a)$; (iii) for each arrow $f \in St$, $Ok(f) = Ok'(f)$; (iv) $\Omega(d) = \Omega(d')$, where d and d' are general arrows of St and St', respectively. In this case we also say that SM' is structurally refined by SM.

Lemma 5.6 The composite of two structural refiners is also a structural refiner.

The lemma above may be used to look for the *most refined system models* and *least refined system models* . All the following specific transformations also have similar lemmas. We will not list them. We can also modify the above definition to have the refined structure of a component of a system.

Definition 5.7 (Behavioural Selection) An isomorphic functor Ω from SM to SM' is said to be a behavioural selector if the following conditions hold: (i) For each object X in $St, Ok(X) \supseteq Ok'(\Omega(X))$. (ii) For any $f \in COMP$, $F(a) = G(a)$, where $F = Ok'(\Omega(f))$ and $G = Ok(f)$.

Encapsulation hides something from us. In practice we may not be interested in all possible inputs or outputs [12].

Definition 5.8 (Encapsulation) An isomorphic functor Ω from SM to SM' is an encapsulator if for each arrow $f \in COMP$ the following condition holds: Let $Ok'(\Omega(f)) = F$ and $Ok(f) = G$. If $F(a) = b$, then there are x_1, x_2, y_1, y_2 such that

$$G(x_1, a, x_2) = \langle y_1, b, y_2 \rangle$$

Definition 5.9 (Simulation) An isomorphic transformation Ω from SM to SM' is said to be a simulation if there is a mapping ϵ such that for every component $f \in COMP$ if $OK(f)(x) = y$ then $OK'(\Omega(f))(\epsilon(x)) = \epsilon(y)$. The mapping ϵ is also called simulator.

In practice simulation is an important technique. For example, in electronic gates we usually describe the behaviour in terms of Boolean values (0 and 1). In factories we may describe the behaviour in terms of electrical voltage. The mapping ϵ may be defined to be the relationship between electrical voltages and Boolean values.

All the above transformations are called *regular transformations*. It turns out that the regular system transformations are very powerful and expressive. For example, Mozetic [12] introduced three refinement operators: refinement/collapse, introduction/deletion and elaboration/simplification. The operator refinement/collapse turns out to be a special case of our simulation, introduction/deletion is a composite of encapsulation and simulation. With respect to elaboration/simplification, the exact relation between two adjacent levels in the hierarchy was not defined, and thus it is not clear how to mathematically compare his elaboration/simplification with our regular transformations. But it seems that the role of elaboration/simplification could be played by composites of structural refinements, simulations and selections.

6 Diagnosis as a process

Diagnosis as a process was proposed by Struss [16, 17]. Subsequent work has focused on formalizing this notion, and most of them on some particular features of the process of diagnosis [1, 7, 5, 2].

In this paper we define the system description SD to be a category. The diagnostic assumptions can be modeled by the morphisms in SD. In order to generate desired diagnosis we may navigate in the diagram of category. Suppose SM_i in SD is picked up as a system model, we then can generate diagnosis candidates (as discussed in Section 4). There may be some morphisms from or to SM_i. Then according to the diagnosis we can use some *strategy* to pick up one of the morphisms to get the next system model. For example, we may adopt such a strategy: Start with the least structurally refined system model, then go to other more structurally refined system models step by step. When working on a system model, we may use the selection, or encapsulation, or simulation to choose a suitable model. The detailed discussion of strategies is out of the scope of this paper, but some specific features may be found in [7]. If there is a functor between two system models, there might be some relationship between their diagnoses, and the relationship can be used to guide the choice of the next model. For the space limitation we will not go into deep discussions, more results may be found in the full version of this paper [10].

Acknowledgements We thank Esprit Project Compulog 2 (No. 6810) and JNICT - Portugal for their support. We would also like to thank José Júlio Alferes, Carlos Damásio, Michael Schroeder and Peter Fröhlich for helpful discussions.

References

[1] Böttcher, C. and Dressler, O., A framework for controlling model-based diagnosis systems with multiple actions, *Annals of Mathematics and Artificial Intelligence*, 11:4, 1994

[2] Damásio, C., Nejdl, W., Pereira, L. M., REVISE: An extended logic programming system for revising knowledge bases, *Proc. of KR94*, 1994

[3] de Kleer, J. and Williams, B.C., Diagnosing multiple faults, *Artificial Intelligence*, 32, 1987, 97-130

[4] Hamscher, W., Modeling digital circuits for troubleshooting, *Artificial Intelligence*, 51, 1991, 223-271

[5] Friedrich, G., Theory diagnosis: A concise characterization of faulty systems, DX'92, 117-125

[6] Friedrich, G., Gottlob, G., Nejdl, W., Phisical Impossibility instead of fault models, AAAI'90, 331–336

[7] Fröhlich, P., Nejdl, W., and Schroeder, M., *A formal semantics for preferences and strategies in modelbased diagnosis*, DX'94, 1994

[8] Greiner, R., Smith, B., and Wilkerson, R., A correction to the algorithm in Reiter's theory of diagnosis, *Artificial Intelligence*, 41:1,1989, 79-88

[9] Hamscher, W., Console, L., and de Kleer, J. (eds.), *Readings in Model-Based Diagnosis*, Morgan Kaufmann, 1992

[10] Li, R. and Pereira, L.M., *Application of Category Theory in Model-Based Diagnostic Reasoning*, Technical Report, UNINOVA, 1994

[11] Mac Lane, S., *Categories for the Working Mathematician*, Springer-Verlag, New York, 1971

[12] Mozetic, I, Hierarchical model-based diagnosis, *Int. J. of Man-Machince Studies*, 35:3, 1991, 329-362

[13] Poole, D., Normality and faults in logic-based diagnosis, IJCAI'89, 1304-1310

[14] Reiter, R., A theory of diagnosis from first principles, *Artificial Intelligence*, 32:1,1987, 57-96

[15] Selman, B. and Levesque, H. J., Abductive and default reasoning: A computational core, AAAI 90, 343-348

[16] Struss, P., Diagnosis as a process, in [9], 1992

[17] Struss, P., What's in SD? Towards a theory of modelling for diagnosis, in [9], 1992

[18] Struss, P. and Dressler, O., Physical negation: Integrating fault models into the general diagnostic engine, IJCAI'89, 1318–1323