Development of Self-Maintenance Photocopiers

Y. Shimomura and **S. Tanigawa**
Mita Industrial Co., Ltd.
Tamatsukuri 1-2-28, Chuo-ku, Osaka, Japan.
Email: {simomura, tanigawa}@mita.co.jp

Y. Umeda and **T. Tomiyama**
Graduate School of Engineering, the University of Tokyo
Hongo 7-3-1, Bunkyo-ku, Tokyo, Japan.
Email: {umeda, tomiyama}@zzz.pe.u-tokyo.ac.jp

Abstract

The traditional reliability design methods are imperfect, because the designed systems aim at less faults but once a fault happens they may *hard-fail.* To solve this problem, we propose the concepts of *self-maintenance machine* (SMM) that can maintain its functions flexibly even though faults happen. In order to achieve capabilities of diagnosing and repair planning, a model based approach that employs qualitative physics is proposed. Regarding repair executing capability, *control type* repair strategy is proposed. A prototype of the self-maintenance machine is developed and it succeeded to maintain its functions as far as its structure does not change. This prototype, however, revealed the following problems that may arise when applied to a commercial product as embedded software. The problems include; (1)performance of the reasoning system, (2)the system size, (3)effects of environmental changes, and (4)roughness of qualitative repair operations. In order to solve these problems, we propose a new reasoning method based on *virtual cases* and *fuzzy qualitative values.* This is a methodology of knowledge compilation that gives better reasoning performance and can deal with real world applications like the self-maintenance machine. By using this method, we finally developed a commercial photocopier that has self-maintenability and is more robust against faults. The commercial version has been supplied worldwide as a product of Mita Industrial Co., Ltd., since April, 1994.

Introduction

While maintenance tasks are hardly automated and need laborious work, it is becoming more crucial that machines do not break or, even if they do, can be repaired quickly because of customers' requirements, costs, and so on. The disasters of Three Mile Island and many aircraft accidents reveal that traditional reliability design method is imperfect, because these systems are designed for less faults but nevertheless once a fault happens they may *hard-fail.* The same problem exists in the maintenance of photocopiers. To reduce such a problem, we should develop *soft-fail* (as opposed to hard-fail) machines that can still maintain their functions even if faults happen.

Although some researchers proposed AI techniques for supporting fault diagnosis (*e.g.* (de Kleer & Williams 1987), (Hamilton 1988), (Milne & Trave-Massuyes 1993)) or technical service (Bell *et al.* 1994), there was few research about automation of diagnosis and repair as an embedded software. However, the photocopier industry has been pursuing such techniques, because photocopiers essentially need 'continuous maintenance and, therefore, maintenance costs including employing service engineers and supplying parts are huge.

In this paper, we first introduce the concept of self-maintenance machines (SMM) (Umeda *et al.* 1994) and its reasoning techniques. Second, we discuss problems in applying these techniques to a commercial machine as an embedded software and propose three new techniques called *virtual cases*, *imitational fault method*, and *job elements* as solutions to these problems. Finally, we illustrate our experience of developing a commercial self-maintenance photocopier to demonstrate the feasibility and industrial relevance of the self-maintenance machine as a design strategy.

Task Description

The ultimate objective of this project is to automate the maintenance of photocopiers. However, it is infeasible or at least very difficult to do so completely with the current technology; for example, consider automation of parts change or cleaning. Instead, we here take an approach to maintain *functionality* of a machine *for a while*, which we call *functional maintenance*, and to wait for a scheduled visit of service man. This approach is useful enough for photocopiers, because the service engineer will visit customers on a regular basis and will maintain the machine in a steady state. For example, suppose a photocopier malfunctions on Friday evening. The service engineer will not (or cannot) come until Monday morning, but some important documents must be copied immediately. The *self-maintenance* photocopier will maintain its function *to*

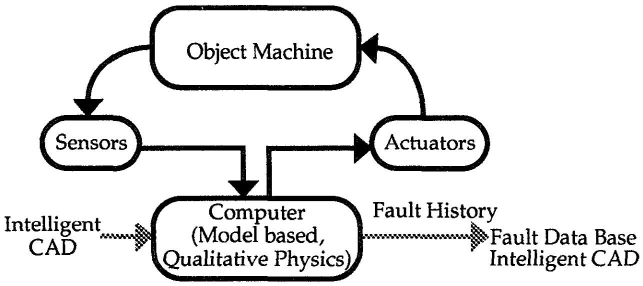


Figure 1: Fundamental Architecture of SMM

make copies by some means, but it might work slower or the copy quality might be slightly worse until the service engineer shows up. The self-maintenance photocopier, therefore, can reduce sudden, unscheduled visits of service engineers, which will greatly reduce the maintenance costs, while minimizing damages to the customers' satisfaction.

One of the features of our approach is that such a self-maintenance photocopier can be developed with existing mechatronics technologies; namely, a photocopier is already equipped with sensors, actuators, and a CPU even for a small to medium-sized analogue photocopier. This suggests that we can build a self-maintenance photocopier just with an additional reasoning system that maintains its functions.

The Framework of SMM

We define a *self-maintenance machine* (SMM) as a machine that can maintain its functions for a while even though faults happen. SMM requires the following five capabilities.

- Monitoring capability
- Fault judging capability
- Diagnosing capability
- Repair planning capability
- Repair executing capability

In order to achieve these capabilities, we developed an architecture for SMM depicted in Figure 1. In this architecture, the monitoring capability and the repair executing capability are executed by sensors and actuators, respectively. The fault judging capability, diagnosing capability, and repair planning capability are obtained through the computer.

One of the key issues of SMM is how to actually execute repairs. Here, we will introduce two types of repairing strategies, i.e., the control type and the functional redundancy type.

- Control Type

  Some repairs of SMM can be accomplished by adjusting parameters without changing or reorganizing the structure. This type of repair is achieved by controlling actuators to recover part or whole of the requested functions, when, e.g., deterioration of a part happens. In some cases, part of the requested functions might be lost or still degraded, but crucial functions will be maintained; this is called *functional trade-off.*

- Functional Redundancy Type

  If a machine has a mechanism that repairs itself by reconfiguring its structure, the range of repairing would be enlarged. For achieving this mechanism, we have proposed a new repair strategy, called *functional redundancy*, which uses potential functions of existing parts in a slightly different way from the original design (Umeda *et al.* 1994). For example, in case of emergency that the engine is broken, a car with a manual transmission can run for a while with a starting motor of which original function is "to start the engine." This *functional redundancy* results from the fact that the starting motor can perform its potential function "to generate driving force" instead of "to start the engine" by changing the power flow of the car.

In the following sections, we mainly discuss the control type SMM.

Prototyping

We developed a prototype that executes the control type repair in order to examine the feasibility of the self-maintenance photocopier (Umeda *et al.* 1994). Figure 2 depicts the prototype in which a photocopier (the object machine) is connected to an MS-DOS[1] based personal computer. The personal computer collects sensory data from the photocopier and outputs control data to it. The personal computer is also connected to a Macintosh[2] computer. The personal computer converts quantitative sensory data to qualitative data and sends them to the Macintosh. Figure 3 shows an example of a quantity space that maps quantitative values to qualitative values and landmarks. The Macintosh reasons about repair plans from the qualitative sensory data with the method described in Section . This reasoning system is implemented on Smalltalk-80[3].

Reasoning Technique for the Prototype. The reasoning system of the prototype is base on a model-based reasoning technique that employs qualitative physics (Weld & de Kleer 1989), because it can solve the following problems that heuristics based expert systems encounter.

- Reusability of knowledge

[1] MS-DOS is a registered trademark of Microsoft Corporation.

[2] Macintosh is a registered trademark of Apple Computer, Inc.

[3] Smalltalk-80 is a registered trademark of ParcPlace Systems, Inc.

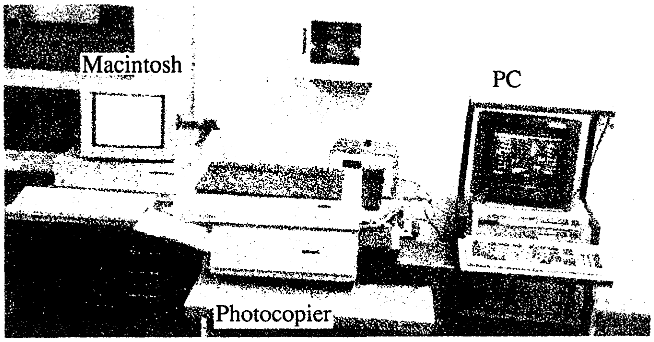


Figure 2: The Prototype of a Self-Maintenance Photocopier

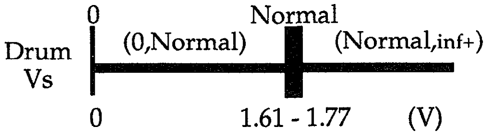


Figure 3: An Example of Quantity Space

Because knowledge dependent on the object machine is localized and isolated in the model, the system can deal with different object machines by changing object models.

- Robustness of the reasoning systems

  The system can deal with unknown cases that are not described explicitly in the heuristics based knowledge by applying generic physical principles to the model.

- Problems of knowledge acquisition

  Since models can be prepared from design information of the object machine and the knowledge about physical principles can be acquired systematically, it is easier to acquire the model based knowledge than the heuristics based knowledge.

Knowledge Representation. The knowledge for the reasoning system consists of a model of the object machine (which we call a *parameter model*) and knowledge about physical principles that describes fault phenomena that might occur on machines in general.

Figure 5 depicts an example of the parameter model of the prototype (Figure 4) that describes physical characteristics of the machine with parameters and their mathematical relations. In this model, *function* parameters are used for measuring functions. For instance, Os denotes the density of the output image on the paper. *Control* parameters signify parameters controllable through actuators. An example of this is HlC that can be controlled by the halogen lamp controller. *Sensing* parameters are monitored by sensors. For example, X should be monitored by the photometer.

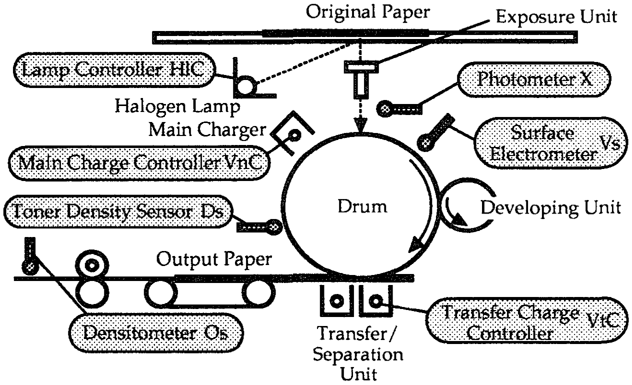


Figure 4: The Structure of the Photocopier

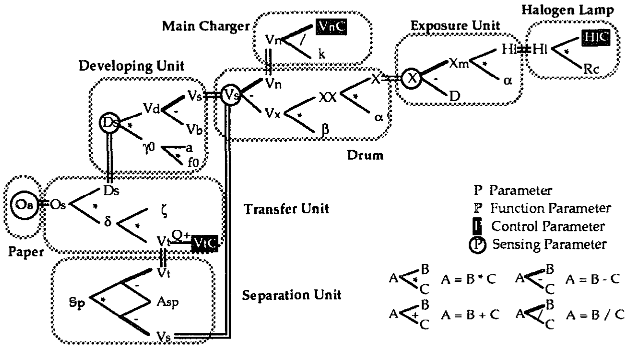


Figure 5: Parameter Model of the Prototype

In this prototype, we have selected two parameters as the function parameters; namely, Os, the density of the output image on the paper and Sp at the separation unit indicating whether or not the output paper comes out from the machine. Since faults might change the structure and/or attributes of the object machine, we describe fault phenomena based on the idea of *process* of Qualitative Process Theory (Forbus 1984). In our approach, every phenomenon is associated with conditions under which this phenomenon occurs and effects it causes (see Figure 6 explained below).

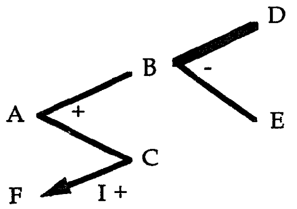
Algorithm of the Reasoning System. The reasoning system of the prototype executes fault judgment, fault diagnosis and repair planning with the algorithm explained below by using an example parameter model depicted in Figure 6.

1. Fault Judgment

First, the system identifies fault symptoms from the sensory data.

2. Fault Diagnosis

1. Search for Fault Cause Candidates



Phenomenon P1
condition: value(F) > Limit1
effect: increase(C)

Phenomenon P2
condition: value(F) < Limit2
effect: break(B = D - E)

Figure 6: Example of the Parameter Model and Phenomena

The qualitative reasoning system searches for all phenomena (called *fault causes*) of which influences are related to the symptom directly or indirectly. In Figure 6 in which functional change is the increase of *A*, phenomena P1 and P2 are selected.

2. Fault Simulation

   Next, the reasoning system carries out qualitative simulation about the behavior of the object machine for each fault cause candidate based on the QSIM algorithm (Kuipers 1986). This simulation determines a faulty state of the object machine (*fault model*) for each fault cause. The fault models provide the repair planning step with information about the structure and the state of the faulty object machine.

3. Identify Fault Candidate

   The reasoning system selects the fault model which matches the most with the sensory data.

3. Repair Planning

The system reasons out a repair plan from the diagnosed fault model. Here, based on the ideas of functional maintenance, we aim at recovering the lost functions (i.e. fault symptoms) rather than fixing the faults themselves (i.e. structural or attributive changes) that are objectives of the traditional repairs. This procedure consists of the following three steps:

1. Generate Repair Candidates

   After the user selects a fault symptom to be recovered, the system derives desired changes of the control parameters from the faulty parameter model. In Figure 6, repair objective is to decrease the value of function parameter *A* and the control parameters are *C* and *E*. We can reason out that decrease of *C* and increase of *E* are the candidates of repair operation.

2. Repair Simulation

   The reasoning system qualitatively simulates the behavior of the object machine when each reasoned repair operation is executed. As a result, the reasoning system examines whether or not a selected repair method can achieve the repair objective and whether or not a selected repair method will cause side effects to the object machine.

3. Select Repair Operation

   The reasoning system selects a repair operation that can repair or improve the objective function and has the least number of side effects.

Examples. Figure 7 illustrates an example operation of the prototype. In this example, the symptom "*Paper Os* is high" and the fault cause "deterioration of the halogen lamp" are identified correctly by the system. Since the reasoning system outputs only qualitative repair operations, the repair executing algorithm includes a heuristic strategy that changes parameter values in a stepwise manner. In this repair procedure, the system first tried to increase the voltage of the lamp controller. However, the controller reaches its control limit without recovering the function because of deterioration of the lamp. Then, the system executed another repair operation in which the main charger controller decreased the main charger voltage and it succeeded in recovering the function.

This example demonstrates both the feasibility and the flexibility of SMM. The latter is achieved, because it can generate repair plans adaptively according to the state of the object machine based on the model based approach.

Problem Description

Based on the techniques described in above section, we have developed a commercial product of self-maintenance photocopier. However, some problems turned out to apply the techniques developed for the prototype system to the commercial machine. In this section, we analyze these problems and propose techniques that solve the problems.

Since the reasoning system of the prototype reasons out all possible faults based on physical principles and executes fault simulation for each diagnosis task, the system has the following features:

1. It can diagnose unexpected faults of the machine including causally related multiple faults (but limited to single cause).
2. It can diagnose faults that change the physical structure of the machine.

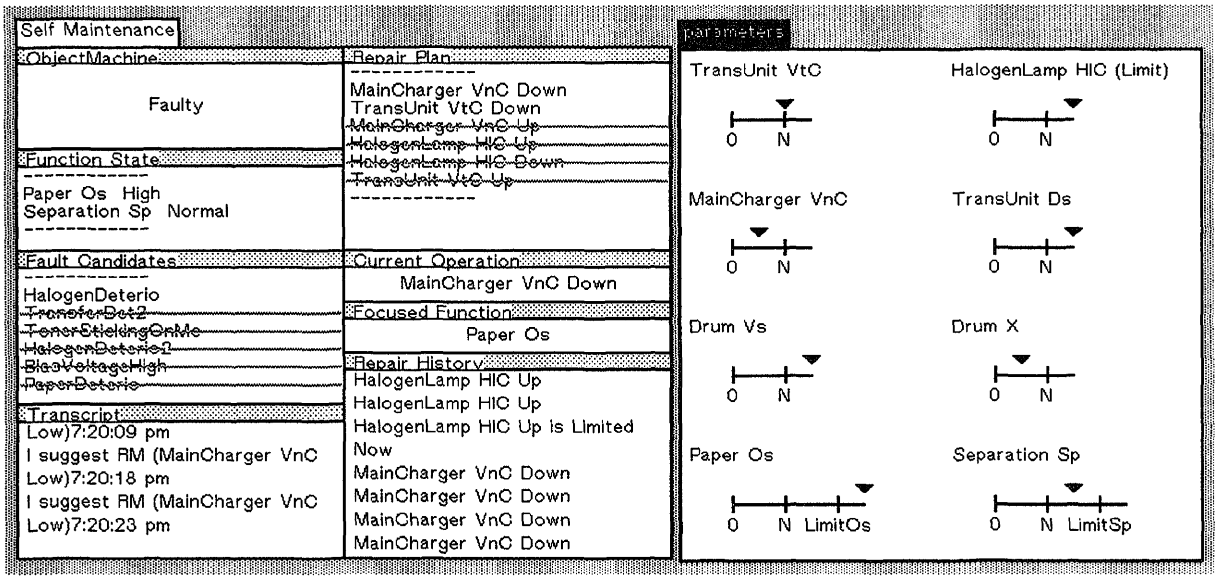
However, to use the system as embedded software of the commercial products, the following problems can be identified:

1. The size of the system and the reasoning speed are unacceptable.

   Since the prototype system is implemented on Smalltalk-80, the size of the system is large as embedded software for commercial products and the reasoning speed is significantly slow.

2. Changes of the environment lead to mistakes in diagnosis.

   In the prototype system, a qualitative space mapping is used for converting quantitative sensory data into qualitative values. However in the real world, it



Self Maintenance

ObjectMachine

Faulty

Function State

Paper Os High
Separation Sp Normal

Fault Candidates

HalogenDeterio
~~TransferDet2~~
~~TonerStickingOnMe~~
~~HalogenDeterio2~~
~~BiasVoltageHigh~~
~~PaperDeterio~~

Transcript

Low)7:20:09 pm
I suggest RM (MainCharger VnC
Low)7:20:18 pm
I suggest RM (MainCharger VnC
Low)7:20:23 pm

Repair Plan

MainCharger VnC Down
TransUnit VtC Down
~~MainCharger VnC Up~~
~~HalogenLamp HIC Up~~
~~HalogenLamp HIC Down~~
~~TransUnit VtC Up~~

Current Operation

MainCharger VnC Down

Focused Function

Paper Os

Repair History

HalogenLamp HIC Up
HalogenLamp HIC Up
HalogenLamp HIC Up is Limited
Now
MainCharger VnC Down
MainCharger VnC Down
MainCharger VnC Down
MainCharger VnC Down

parameters

Figure 7: Screen Hardcopy of the Reasoning System

is not easy to obtain the mapping statically because of imprecision of the sensors and changes of the environment around the machine. Incorrect mapping results in mistakes in the fault diagnosis.

3. Roughness of the qualitative repair operation.
   Since the prototype system is based on the qualitative reasoning, the system can deal with faults flexibly. On the other hand, the qualitative representation of states of the machine is too rough to judge whether an repair operation is correct or not.

For solving these problems, we have developed the following new techniques that will be discussed below.

- *Virtual case* based reasoning for problem 1.
- The *imitational fault method* for problem 2.
- *Division of the quantity space* and the *job elements* for problem 3.

Application Description

Virtual Case

To solve the problems about the reasoning speed and system size, we adopt the single fault assumption, and compile all possible states of the machine including faulty states as compiled knowledge. For this purpose, we have developed a tool called a *virtual case compiler.* A virtual case is a set of qualitative values of the parameters of the parameter model and represents a state of the object machine. The virtual case compiler generates all possible virtual cases through qualitative simulation of the prototype reasoning system over all faults and all operations of actuators. One of the advantages of this method is that the compiler uses the qualitative simulator without any modifications. It also means that the qualitative simulator can be separated from the diagnostic engine. As a result, we can reduce the size of embedded software and the reasoning speed increases, because this makes the reasoning algorithm much simpler.

Table 1: The Virtual Cases

Cases	Sensory Parameters			Qualitative States	Fault	Symptom
	Ds	Vs	X			
a	+	+	−	Hl = -, D = N, X = -, Beta = N, ···	Hl decrease	fog image
				Hl = -, D = N, X = -, Beta = N, ···		
				⋮		
b	+	+	N	Hl = N, D = N, X = N, ···	Vn increase	fog image
				Hl = N, D = N, X = N, ···		
				⋮		
⋮		⋮		⋮	⋮	⋮

N: normal value
+: larger than normal
-: smaller than normal

Table 1 shows examples of the virtual cases that are compiled from the parameter model shown in Figure 5. In Table 1, a virtual case "a" that can be identified by the sensory values " Ds=+", "Vs=+",and "X=-" corresponds to a set of qualitative states that includes faulty states.

Imitational Fault Method

A qualitative space mapping is needed for converting quantitative sensory data into qualitative values. However, due to errors and instability of the sensors and changes of the environment around the machine, it is difficult to obtain this mapping of which accuracy is fa-

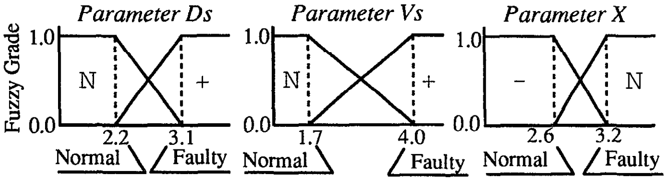


Figure 8: Fuzzy Qualitative Space Map

tal to the correctness of the fault diagnosis. In order to solve this problem, we have developed two techniques; namely, the *fuzzification of qualitative value* (Umeda, Tomiyama, & Yoshikawa 1992) and the *imitational fault method.* The fuzzification of qualitative value is a method to relate numerical sensory data to qualitative values by means of fuzzy theory (Zadeh 1965).

The imitational fault method calibrates the mapping between sensory values and fuzzy qualitative values dynamically by checking up the fuzzy membership functions through imitationally caused faults every time a fault repair operation is executed. The imitational fault method is executed in the following steps:

1. After a repair operation, the system causes faults intentionally by controlling an actuator until a symptom occurs. The actuator that may change values of all the sensors is appropriate for this. Currently, the designer selects the actuator to cause imitational faults.
2. The system revises the fuzzy qualitative space mapping for all sensory parameters of each symptom to distinguish faulty and normal sensory data.

Figure 8 shows examples of the fuzzy qualitative space mappings generated by this method when a symptom "the increase of parameter Os" is imitationally caused by the decrease of the halogen lamp controller that is an actuator for *Hl* (see Figure 5). For example, the mapping for the parameter *Ds* in Figure 8 signifies that the symptom definitely appears and disappears when its sensory value reads more than 3.1 and less than 2.2, respectively, but it is not obvious to conclude that the symptom appears when the value is between 2.2 and 3.1.

Division of the Quantity Space

As described later, virtual cases are ordered according to their matching ratios to the machine's state as the result of diagnosis. This order of cases identifies the most probable candidate of fault and divides the quantity space of the object machine into some areas in the following way. Here, the quantity space is defined as a vector space of all sensory parameters of which dimension is the number of the sensory parameters in the machine. The order of cases can be considered equivalent to the order of distances to the cases of the current machine's state.

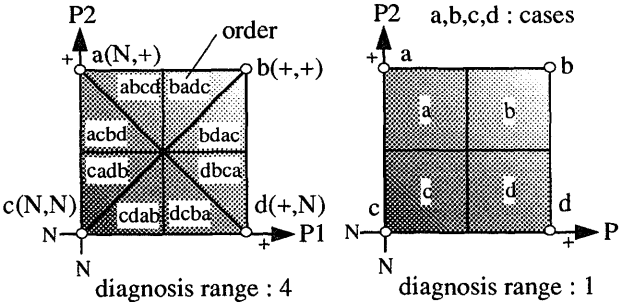


Figure 9: Example of Qualitative Space Division

Figure 9 depicts an example of this divided space that consists of two sensory parameters P1 and P2 and, therefore, that has four cases, viz., a(N, +), b(+, +), c(N, N), and d(+, N), if only the positive regions of the parameters are considered. These four cases can be ordered according to their matching ratios. For instance, with a certain set of the sensor values, case b is the most probable and case d is the second most. This order, bd, corresponds to an area which is a division of this quantity space. We may consider a certain number of cases in this ordering. This number is called *diagnosis range* and, thus, determines the number of the division of the space. For example, Figures 9 (A) and (B) show the spaces of which diagnosis ranges are four and one, respectively. The number of the divided areas in the quantity space affects repair accuracy, we should decide this number experimentally with respect to the diagnosis range. The diagnosis range should be large enough to execute the repair operations precisely, but it should not be too large, because the divided quantity space becomes too precise to be detected correctly by the sensors. Currently, the diagnosis range of the commercial product is experimentally determined.

The maximum number of divided areas of the quantity space, *Nmax*, is given by the following equation, where *n* denotes the number of sensors.

$$Nmax = 2^n \times {}_n\mathrm{P}_2 \times \prod_{r=2}^{n-1}((n-r)+\sum_{k=2}^{r}{}_r\mathrm{C}_k) \quad (n \geq 3) \quad (1)$$

Table 2 shows the result of area division of a quantity space consisting of three sensors, and the space is divided into 96 areas using the maximum diagnosis range of eight. In Table 2, symbols such as a and b denote virtual cases and each area corresponds to an order of these virtual cases according to their matching ratios.

Job Elements

Each repair operation through controlling an actuator means a transition of areas in the divided quantity space. However, there are possible and impossible

Table 2: Area Dividing of a Quantity Space of Three Sensors

Area Name	Order of Cases	Area Name	Order of Cases
A-1	abdcefhg	A-49	eafbhdgc
A-2	abdcefhg	A-50	eafhbdgc
⇓	⇓	⇓	⇓
A-47	dhcagebf	A-95	hgedfcab
A-48	dhcgaebf	A-96	hgefdcab

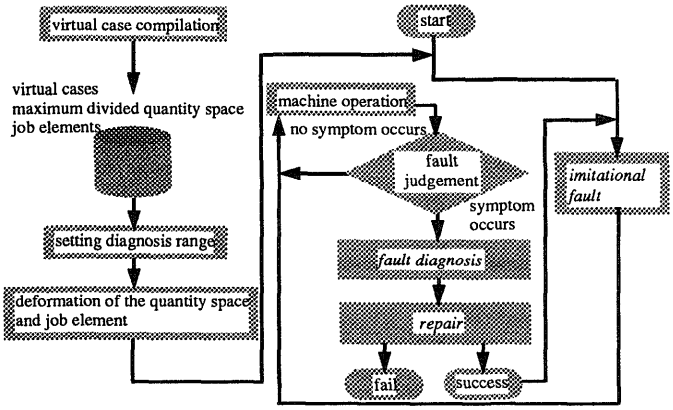


Figure 10: The Reasoning Algorithm of the Commercial Photocopier

transitions from one area to neighboring areas because of physical characteristics of the object machine. Although these transition rules for the commercial product are experimentally obtained, future work should aim at deriving these rules automatically from the parameter model. We call this rule a *job element* that represents the relation between an operation of an actuator and area transition. Table 3 shows examples of job elements. The job elements are described for each area in the divided quantity space.

Reasoning Algorithm of the Commercial Photocopier

In this section, we explain the reasoning algorithm of the commercial photocopier. Figure 10 represents the reasoning algorithm.

Models and Knowledge. During the development process, first the following knowledge for each kind of machine must be prepared.

1. The parameter model set up manually and virtual cases compiled from the parameter model with the virtual case compiler.
2. The maximumly divided quantity space and job elements.

Then, the following data must be determined.

3. The diagnosis range.
4. The divided quantity space and the job elements modified with respect to the diagnosis range.
5. The fuzzy qualitative space map for each sensory parameter obtained through the imitational fault method.

Diagnosis. The commercial version of self-maintenance photocopier diagnoses itself based on virtual cases by searching for a virtual case that has the highest matching ratio with the sensory data of the object machine. The matching ratio is calculated by comparing the fuzzy qualitative values of sensors with the qualitative values in each virtual case as given in equations (2) and (3).

$$C = 1 - \sqrt{C(p_1)^2 + C(p_2)^2 + \cdots + C(p_n)^2}/\sqrt{n} \quad (2)$$
$$C(p_i) = Gm(q_i) - Gs(q_i) \qquad (i = 1, 2, \cdots, n) \quad (3)$$

where
C: the matching ratio
i: the number of a sensor
p_i: a sensory parameter
q_i: a qualitative value
Gm: a fuzzy value of q_i in a virtual case
Gs: a sensed fuzzy value of q_i

Table 4 lists ordered virtual cases as an example of the fault diagnosis for the following fuzzy qualitative data when the symptom "increase of parameter Os" occurs. In this data, the value of parameter Ds, for instance, indicates that the fuzzy grade of '+' (larger than normal) is 0.6 and the fuzzy grade of 'N' (normal) is 0.4 (see Figure 8).

$$Ds(+ : 0.6, N : 0.4) \quad (4)$$
$$Vs(+ : 0.6, N : 0.4) \quad (5)$$
$$X(- : 0.8, N : 0.2) \quad (6)$$

Fault diagnosis, thus, identifies an area in the quantity space of the object machine. Figure 11 shows the algorithm of the fault diagnosis based on the combination of virtual cases and the imitational fault method.

Repair. After fault diagnosis, the system executes repair planning and a repair operation with the job elements as follows:

1. The system searches for a path from the current area to the goal area by combining job elements. It is

Table 3: The Job Elements

Job Element		Area : A-34	
Operation	**Next Areas**	**Operation**	**Next Areas**
Hl up	A-36	Hl down	A-33, A-28, A-27
Vn up	A-28, A-33, A-27	Vn down	A-36
Vb up	A-36	Vb down	A-33

Table 4: Example of Fault Diagnosis

Virtual Case					Matching Ratio			
Order	**Fault**	**Ds**	**Vs**	**X**	**C(Ds)**	**C(Vs)**	**C(X)**	**C**
1	Hl control	+1.0	+1.0	-1.0	0.4	0.4	0.2	0.65
2	Vn control	+1.0	+1.0	N1.0	0.4	0.4	0.8	0.43
3	Vb control	+1.0	N1.0	N1.0	0.4	0.6	0.8	0.38
4	Vt control	N1.0	N1.0	N1.0	0.6	0.6	0.8	0.33
⇓	⇓		⇓			⇓		⇓

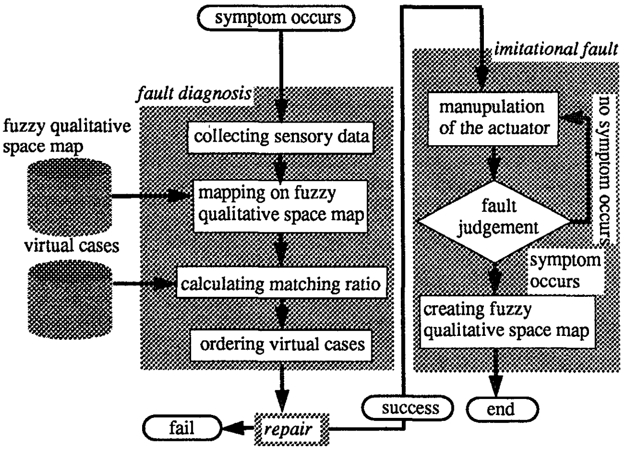


Figure 11: The Algorithm of Fault Diagnosis

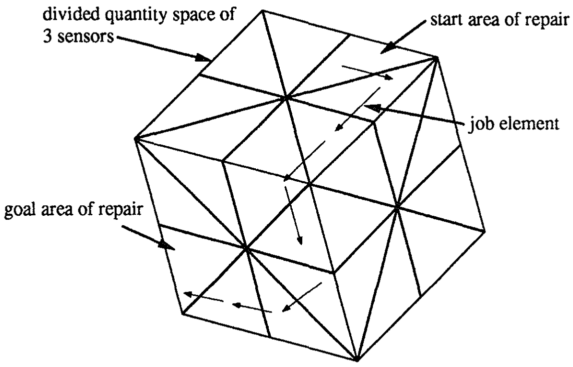


Figure 12: The Concept of Repair with Job Elements

similar to techniques which have been developed in software analysis in respect of using for all execution paths that some code might take (Srimani & Malaiya 1992).

2. It manipulates the actuators in order to change the state of the object machine along the path, continuously comparing the current state with the supposed-to-be states described in the 'Next Areas' of the current job element

3. If the current state of the machine is different from the supposed-to-be state, the system stops the operation immediately and restores the state to the initial one. In such a case, the diagnosis range of the machine might have changed by, for example, deterioration of sensors.

Figures 12 and 13 show the concept of repair with job elements and the repairing algorithm, respectively.

Executing the Imitational Fault Method. After executing repair, the system modifies the fuzzy qualitative space map of each sensory parameter by executing the imitational fault method.

Application Use and Payoff

We have developed a commercial photocopiers based on the techniques described so far. Figure 14 and 15 depicts its architecture and exterior. Figure 16 is a screen hardcopy of the virtual case compiler used for the development. Figure 17 compares a faulty output image and a repaired one. The inference system of the commercial product was developed on the C and assembler languages, and takes at most 10 minutes for a diagnosis and a repair including revision of the fuzzy qualitative space mapping. The system size is about 280 Kbytes including 16 virtual cases and its diagnosis range is 2. These results are acceptable as an embed-

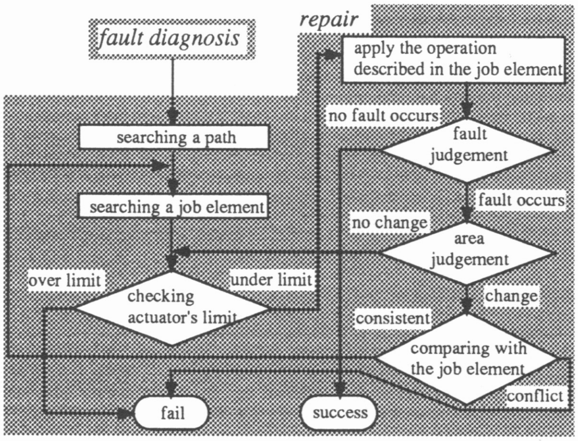


Figure 13: The Algorithm of the Fault Repairing

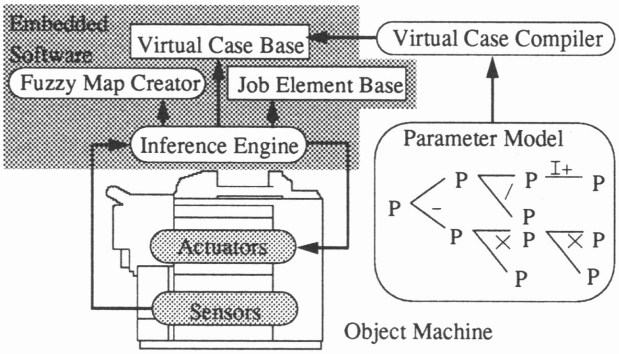


Figure 14: The Structure of the Commercial System

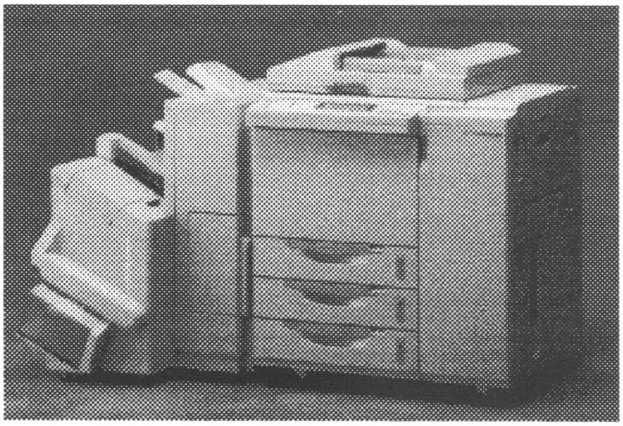
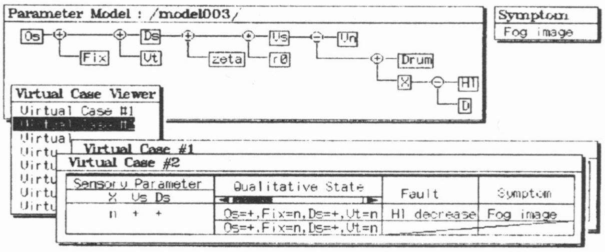


Figure 15: The Exterior of the Commercial Machine



Virtual Case #2

Sensory Parameter X Us Ds	Qualitative State	Fault	Symptom
n + +	Os=+,Fix=n,Ds=+,Ut=n	Hl decrease	Fog image
	Os=+,Fix=n,Ds=+,Ut=n		

Figure 16: An Screen Hardcopy of the Virtual Case Compiler

ded system. Figure 18 shows typical results of some experimental diagnoses that demonstrates the diagnosis against fuzziness of the sensory information.

While the knowledge compilation techniques based on the virtual case greatly improves the reasoning speed and the system size, the system has less flexibility against faults with structural changes compared with simulation based systems.

The commercial version photocopier has been supplied to the market world wide since April, 1994. Although it is too early to evaluate payoff of the product, we predict the decreases of down-time and maintenance costs paying off the cost of the development. At least, service engineers, particularly in the US, prefer this self-maintenance photocopier to traditional ones, because it can reduce their work while maintain the level of service.

Application Development and Deployment

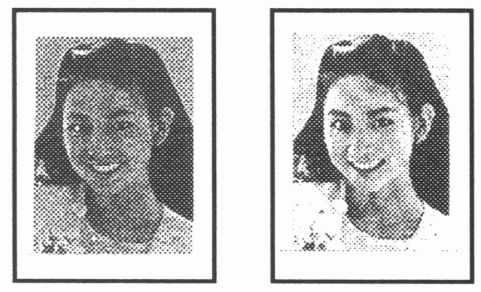
The commercial system was developed by 10 designers over two years following the development of the prototype system in the "self-maintenance machine" project at the University of Tokyo in 1992. The design team includes mechanical, electrical, and software engineers. The designers spent the most of the time of development on experiments for the sensors and software development and they also have started to deploy these techniques to other products.

Maintenance

We have not yet found out any necessity of maintenance of the knowledge or the reasoning algorithm of the system.

Conclusion

In this paper, we have proposed a design strategy for the self-maintenance machine and its prototype. We then proposed a method to reduce the system size and to improve the performance of reasoning, combined with a method to treat sensory data with fuzzy qualitative reasoning and imitational fault method. We also proposed a method to execute repair operations more accurately. The development of a commercial version of the self-maintenance photocopier demonstrated its feasibility.



Faulty Image Repaired Image

Figure 17: The Result of Fault Repairing

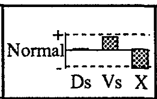
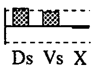
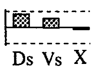
Quantitative Sensory Data				
Qualitative Value	Fuzzy	(+0.1,+0.9,-1.0)	(+1.0,+0.9,-0.1)	(+0.8,+0.6,-0.2)
	Non Fuzzy	(N,N,-)	(+,N,N)	(N,N,N)
Diagnosis Results	Fuzzy	○ halogen lamp	○ main charge	○ main charge
	Non Fuzzy	× transfer	× developing bias	× transfer

Figure 18: The Result of the Fault Diagnosis

Future issues include:

- Making repair planning more flexible by allowing users to select repair goals based on the *functional trade-off* (Umeda *et al.* 1994).
- Developing a machine that can treat faults with structural changes based on the concepts of *functional redundancy* (Umeda *et al.* 1994).

References

Bell, D.; Bobrow, D.; Falkenhainer, B.; Fromherz, M.; Saraswat, V.; and Shirley, M. 1994. Rapper: The copier modeling project. In Nishida, T., ed., *Proceedings of 8th International Workshop on Qualitative Reasoning about Physical Systems*, 1–12.

de Kleer, J., and Williams, B. C. 1987. Diagnosing multiple faults. *Artificial Intelligence* 32(1):97–130.

Forbus, K. 1984. Qualitative process theory. *Artificial Intelligence* 24(3):85–168.

Hamilton, T. P. 1988. Helix: A helicopter diagnostic system based on qualitative physics. *Artificial Intelligence in Engineering* 3(3):141–150.

Kuipers, B. 1986. Qualitative simulation. *Artificial Intelligence* 29(3):289–338.

Milne, R., and Trave-Massuyes, L. 1993. Real-time model based diagnosis of gas turbines. In *Proceedings of IJCAI-93 Workshop – Engineering Problems for Qualitative Reasoning –*, 49–57.

Srimani, P., and Malaiya, Y. 1992. Steps to practical reliability measurement. *IEEE Software* 9(4):10 – 12.

Umeda, Y.; Shimomura, Y.; Tomiyama, T.; and Yoshikawa, H. 1994. Using functional maintenance to improve fault tolerance. *IEEE Expert* 2(3):25–31.

Umeda, Y.; Tomiyama, T.; and Yoshikawa, H. 1992. A design methodology for a self-maintenance machine. In *First International Conference on Intelligent Systems Engineering*, number 360 in Conference Publication, 179–184. The Institution of Electrical Engineers (IEE).

Weld, D., and de Kleer, J., eds. 1989. *Readings in Qualitative Reasoning about Physical Systems.* San Mateo, CA: Morgan-Kaufmann.

Zadeh, L. A. 1965. Fuzzy sets. *Information Control* 8:338–353.