AI Magazine Volume 16 Number 4 (1995) (© AAAI)

Development of Self-Maintenance Photocopiers

Yoshiki Shimomura, Sadao Tanigawa, Yasushi Umeda, and Tetsuo Tomiyama

■ The traditional reliability design methods are imperfect because the designed systems aim at fewer faults, but once a fault happens, the systems might hard fail. To solve this problem, we present a *self-maintenance machine* (*SMM*), one that can maintain its functions flexibly even though faults occur. To achieve the capabilities of diagnosing and repair planning, a model-based approach that uses qualitative physics was proposed. Regarding the repair-executing capability, a control-type repair strategy was followed. A prototype of the SMM was developed, and it succeeded in maintaining its functions if the structure did not change. However, the prototype revealed the following problems when its reasoning system was used with a commercial product as embedded software: (1) poor performance of the reasoning system, (2) system size that was too large, (3) low adaptability to environmental changes, and (4) roughness of qualitative repair operations. To solve these problems, we proposed a new reasoning method based on virtual cases and fuzzy qualitative values. This methodology is one of knowledge compilation, which gives better reasoning performance and can deal with real-world applications such as the SMM. By using this method, we finally developed a commercial photocopier that has self-maintainability and is more robust against faults. The commercial version has been supplied worldwide as a product of Mita Industrial Co., Ltd., since April 1994.

Maintenance tasks are hardly automated and require laborious work, and it is becoming more crucial, because of customers' requirements, costs, and so on, that machines do not break or, even if they do, that they can be repaired quickly. The disaster at Three Mile Island and many aircraft accidents reveal that the traditional reliability design method is imperfect: Although systems based on these design methods are created to handle fewer faults, once a fault happens, the systems can hard fail. The same problem exists in the maintenance of photocopiers. To reduce the problem, we should develop *soft-fail* (as opposed to hard-fail) machines that can still maintain their functions even if faults occur.

Although some researchers have proposed AI techniques for supporting fault diagnosis (for example, Hamilton [1988], de Kleer and Williams [1987], and Milne and Trave-Massuyes [1993]) or technical service (Bell et al. 1994), there has been little research into the automation of diagnosis and repair as embedded software. The photocopier industry has been pursuing such techniques because photocopiers essentially need continuous maintenance, and maintenance costs, including employing service engineers and supplying parts, are huge.

In this article, we first introduce the concept of the self-maintenance machine (SMM) (Umeda et al. 1994) and its reasoning techniques. Second, we discuss problems in applying these techniques to a commercial machine with embedded software and propose three new techniques—(1) virtual cases, (2) imitational fault method, and (3) job elements—as solutions to these problems. Finally, we describe our experience with developing a commercial self-maintenance photocopier to demonstrate the feasibility and industrial relevance of the SMM as a design strategy.

Copyright © 1995, American Association for Artificial Intelligence. All rights reserved. 0738-4602-1995 / $2.00

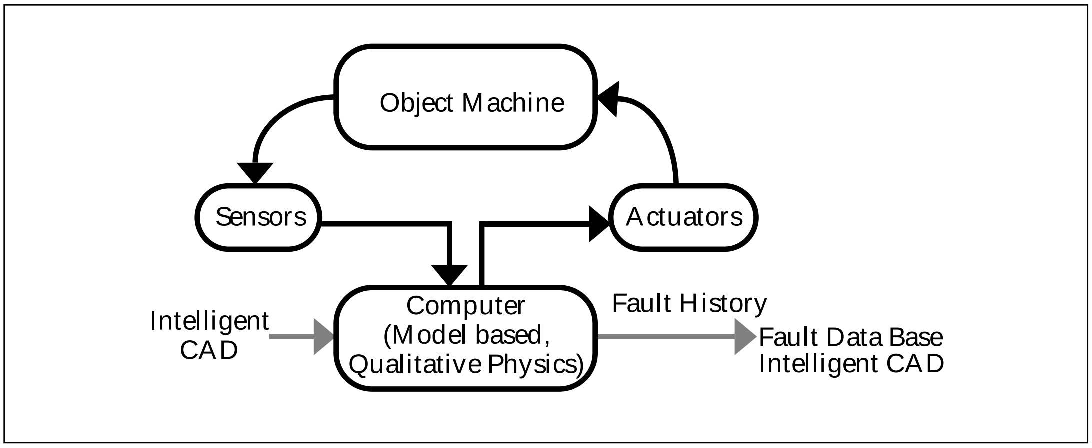

Object Machine
Sensors
Actuators
Intelligent CAD
Computer (Model based, Qualitative Physics)
Fault History
Fault Data Base Intelligent CAD

Figure 1. Fundamental Architecture of an SMM.

Task Description

The ultimate objective of this project was to automate the maintenance of photocopiers. However, it was infeasible or at least very difficult to do so completely using the current technology; consider, for example, the automation of a part change or a cleaning. We took a different approach—maintain the functions of a machine for a while, what we call *functional maintenance*, and wait for the scheduled visit of the service person. This approach is useful 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 a 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 and 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 the need for sudden, unscheduled visits of service engineers, which greatly reduces the maintenance costs and minimizes the damages, all to the customers' satisfaction.

We proposed that a self-maintenance photocopier could be developed with existing mechatronics technologies. A photocopier, even a small- to medium-sized analog photocopier, is already equipped with sensors, actuators, and a central-processing unit. This arrangement suggested that we could build a self-maintenance photocopier with just an additional reasoning system to maintain its functions.

The Framework of the Self-Maintenance Machine

We define an SMM as a machine that can maintain its functions for a while even though faults occur. An SMM requires the following five capabilities: (1) monitoring, (2) fault judging, (3) diagnosis, (4) repair planning, and (5) repair execution. To achieve these capabilities, we developed an architecture for an SMM, depicted in figure 1. In this architecture, the monitoring capability and the repair-execution capability are executed by sensors and actuators, respectively. The fault-judging, diagnosis, and repair-planning capabilities are obtained through the computer.

One of the key issues of an SMM is how to actually execute repairs. Here, we introduce two types of repair strategy: (1) control and (2) functional redundancy.

Control-Type Repair Strategy Some repairs of an 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 all of the requested functions, for example, when a part deteriorates.

Functional Redundancy–Type Repair Strategy If a machine has a mechanism that repairs itself by reconfiguring its structure, the range of possible repairs would be enlarged. To create this mechanism, we proposed a new repair strategy, called *functional redundancy*, which uses potential functions of existing parts in a slightly different way than intended in the original design (Umeda et al. 1994). For example, with a broken engine, a car with a manual transmission can run for a while with a starting motor that has the original function "to start the engine." Functional redundancy results when the starting motor can perform

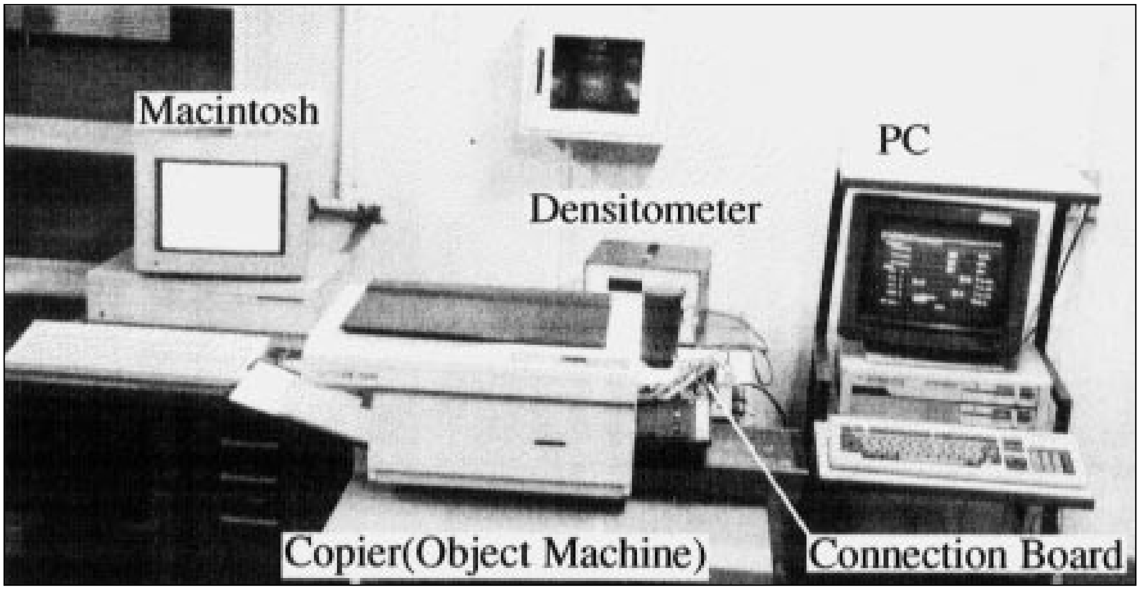


Figure 2. The Prototype of a Self-Maintenance Photocopier.

its potential function "to generate driving force" instead of "to start the engine" by changing the power flow of the car.

In the following subsections, we primarily discuss the control-type SMM.

Prototyping

We developed a prototype that executes the control-type repair 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–based PC. The PC collects sensory data from the photocopier and outputs control data to it. The PC is also connected to a MACINTOSH computer. The PC converts quantitative sensory data to qualitative data and sends them to the MACINTOSH. Figure 3 illustrates a *qualitative space map,* which maps quantitative values to qualitative values. The MACINTOSH reasons about repair plans from the qualitative sensory data with the method described in the next subsection. This reasoning system is implemented on SMALLTALK-80.

Reasoning Technique for the Prototype

The prototype's reasoning system is based on a model-based–reasoning technique that uses qualitative physics (Weld and de Kleer 1989); the system can solve the following problems that heuristic-based expert systems encounter:

Reusability of knowledge: Because knowledge that is 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 heuristic-based knowledge by applying generic physical principles to the model.

Problems of knowledge acquisition: Because models can be prepared from design information from the object machine and the knowledge about physical principles can be acquired systematically, it is easier to acquire the model-based knowledge than the heuristic-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 4 presents an example of the parameter model of the prototype (figure 5) that describes physical characteristics of the machine using parameters and their mathematical relations. In this model, *function parameters* are used for measuring functions. For example, *Os* denotes the density of the output image on the paper. *Control parameters* signify parameters controllable through actuators. An example of this is *HlC,* which can be controlled by the halogen lamp controller. *Sensing parameters* are monitored by sensors. For example, *X* should be monitored by the

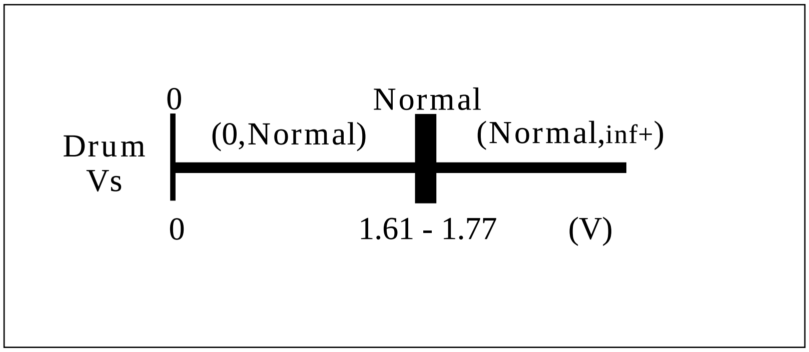


Figure 3. An Example of Qualitative Space Map.

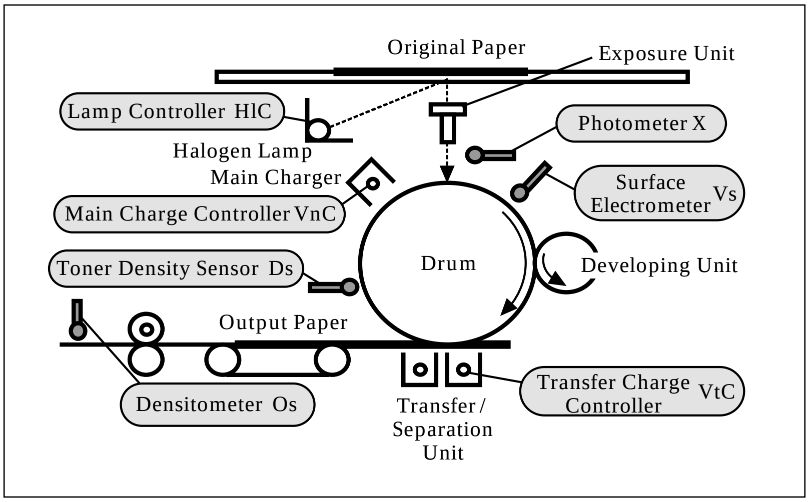


Figure 4. The Structure of the Photocopier.

photometer. In this prototype, we selected two parameters as the function parameters, that is, *Os*, the density of the output image on the paper, and *Sp,* the indication at the separation unit of whether the output paper comes out of the machine.

Because faults might change the structure or attributes of the object machine, we describe fault phenomena based on the idea of qualitative process theory (Forbus 1984). In our approach, every phenomenon is associated with conditions under which this phenomenon occurs and the effects it causes (figure 6).

Algorithm of the Reasoning System

The reasoning system of the prototype executes fault judgment, fault diagnosis, and repair planning with the algorithm. We explain it here using an example parameter model (figure 6).

1. Fault judgment: First, the system identifies fault symptoms from the sensory data.

2. Fault diagnosis: The three steps in fault diagnosis are (1) the search for fault-cause candidates, (2) fault simulation, and (3) the identification of the fault candidate.

1. Search for fault-cause candidates: The qualitative reasoning system searches for all phenomena (*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 *P*1 and *P*2 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. Identification of fault candidate: The reasoning system selects the fault model that 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 (that is, fault symptoms) rather than fixing the faults themselves (that is, structural or attributive changes) that are objectives of the traditional repairs. This procedure consists of three steps: (1) generate repair candidates, (2) simulate repair, and (3) select repair operation.

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

2. Simulate repair: 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 a selected repair method can achieve the repair objective and whether a selected repair operation 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. Because the reasoning system outputs only qualitative

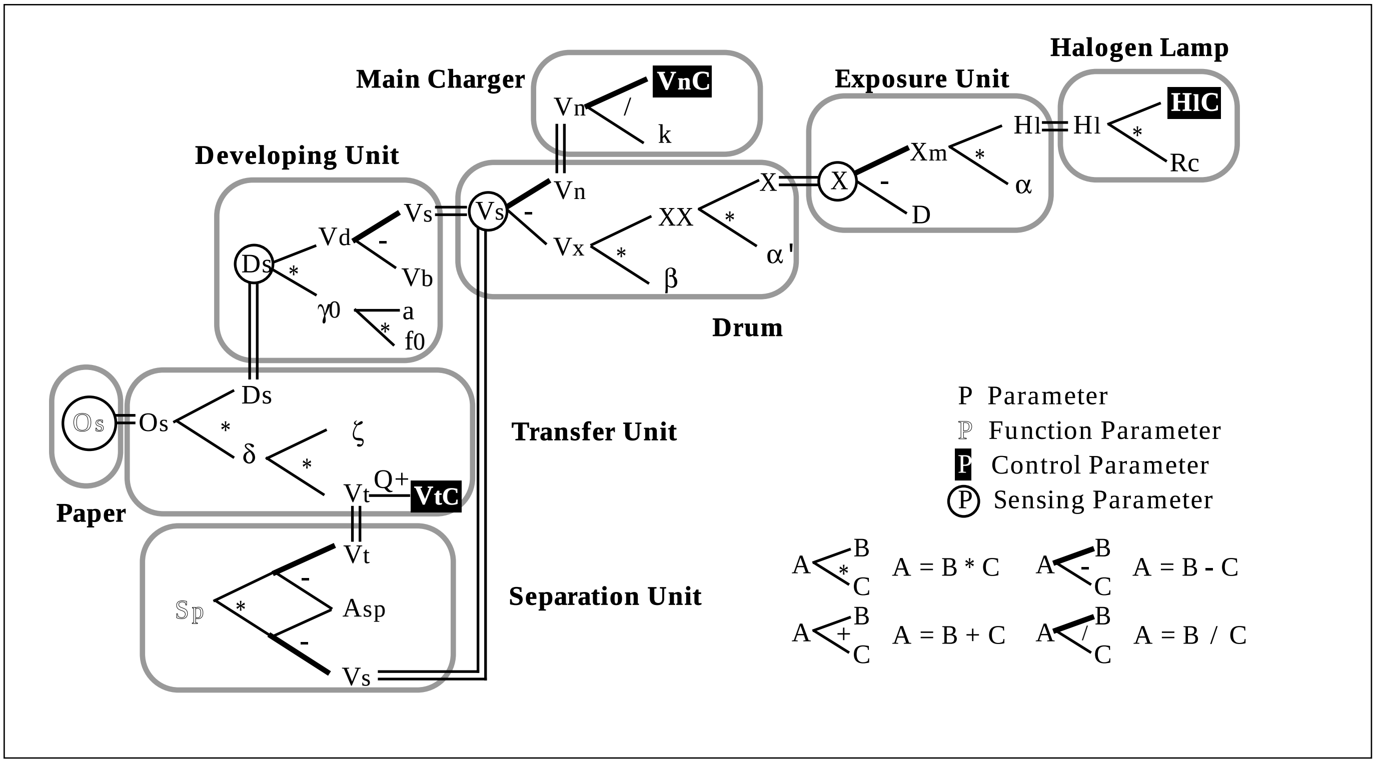
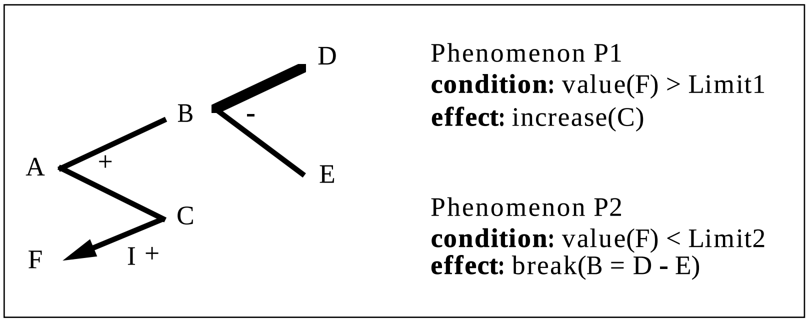


Figure 5. Parameter Model of the Prototype.



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.

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 reached 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 the SMM. The flexibility is achieved because, based on the model-based approach, the system can generate repair plans adaptively according to the state of the object machine.

Problem Description

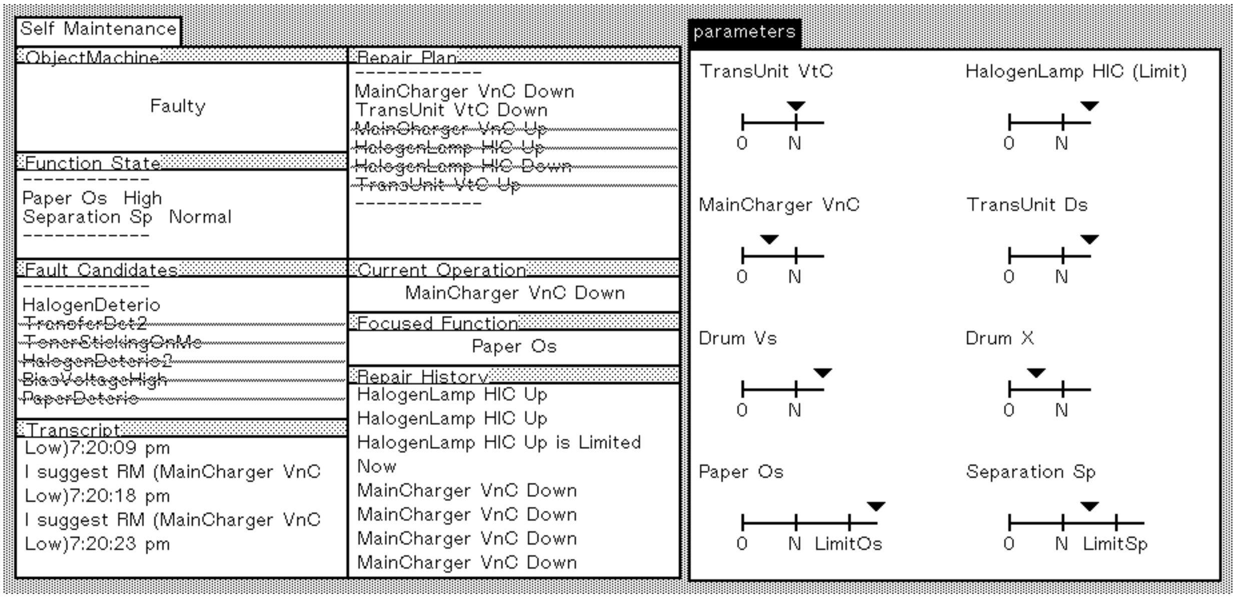
Based on the techniques described in the previous section, we developed a commercial self-maintenance photocopier. However, some problems appeared when applying the techniques developed for the prototype system to the commercial machine. In this section, we analyze these problems.

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), and (2) it can diagnose faults that change the physical structure of the machine.

However, to use the system as embedded software within the commercial products, we identified the following problems:

First, the size of the system and the reasoning speed were unacceptable. Because the prototype reasoning system was implemented on SMALLTALK-80, the size of the system was large, as the embedded software, and the reasoning speed was significantly slow.

Second, changes in the environment led to mistakes in diagnosis. In the prototype sys-



Self Maintenance

ObjectMachine

Faulty

Function State

Paper Os High
Separation Sp Normal

Fault Candidates

HalogenDeterio
~~TransferDet2~~
~~TonerStickingOnMc~~
~~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

Figure 7. Screen Image of the Reasoning System.

tem, a qualitative space mapping is used to convert quantitative sensory data into qualitative values. However, in the real world, it is not easy to obtain the mapping statically because of the imprecision of the sensors and changes in the environment around the machine. Incorrect mapping results in mistakes in the fault diagnosis.

Third, the qualitative repair operation was rough. Because the prototype system was based on qualitative reasoning, the system could deal with faults flexibly. However, the qualitative representation of states of the machine was too rough to judge whether a repair operation was correct.

Application Description

To solve the problems outlined in the previous section, we developed three new techniques: (1) virtual case-based reasoning, (2) imitational fault method, and (3) division of the quantity space and the job elements. These techniques are described in the following subsections.

Virtual Case-Based Reasoning

To solve the problems of reasoning speed and system size, we adopted the single-fault assumption and compiled all possible states of the machine, including faulty states, as compiled knowledge. For this purpose, we 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 advantage 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 embedded diagnostic engine. As a result, we can reduce the size of the embedded software, thus increasing reasoning speed and making the reasoning algorithm much simpler.

Table 1 shows examples of the virtual cases that were 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 to convert quantitative sensory data into qualitative values. However, because of errors, the instability of the sensors, and changes in the environment around the machine, it is difficult to obtain this mapping, where accuracy is fatal

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. The Virtual Cases.

to the correctness of the fault diagnosis. To solve this problem, we developed two techniques: (1) the "fuzzification" of qualitative value (Umeda, Tomiyama, and Yoshikawa 1992) and (2) the imitational fault method.

The *fuzzification of qualitative value* is a method to relate numeric sensory data to qualitative values by means of fuzzy theory (Zadeh 1965).

The *imitational fault method* dynamically calibrates the mapping between sensory values and fuzzy qualitative values dynamically by checking on the fuzzy membership functions through imitationally caused faults every time a fault repair operation is executed. The imitational fault method is executed in two steps: First, after a repair operation, the system causes faults intentionally by controlling an actuator until a symptom occurs. The actuator that can change values of all the sensors is appropriate for this action. Currently, the designer selects the actuator to cause imitational faults. Second, 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 the 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. However, it is not obvious to conclude that the symptom appears when the value is between 2.2 and 3.1.

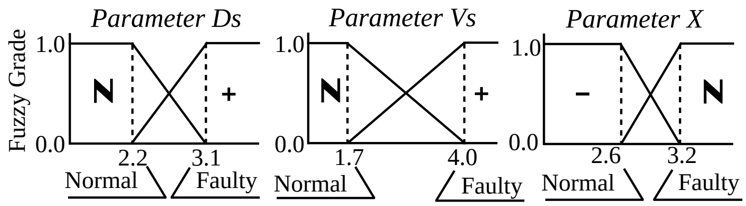


Figure 8. Fuzzy Qualitative Space Map.

Division of the Quantity Space

As described later, virtual cases are ordered according to the ratios that match the machine's state as a result of diagnosis. This order of cases identifies the most probable candidate for 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 from the current machine's state.

Figure 9 presents an example of this divided space that consists of two sensory parameters *P*1 and *P*2 and, therefore, has four cases—*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 example, with a certain set of sensor values, case *b* is the most probable, and case *d* is the second-most proba-

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

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

	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 3. The Job Elements.

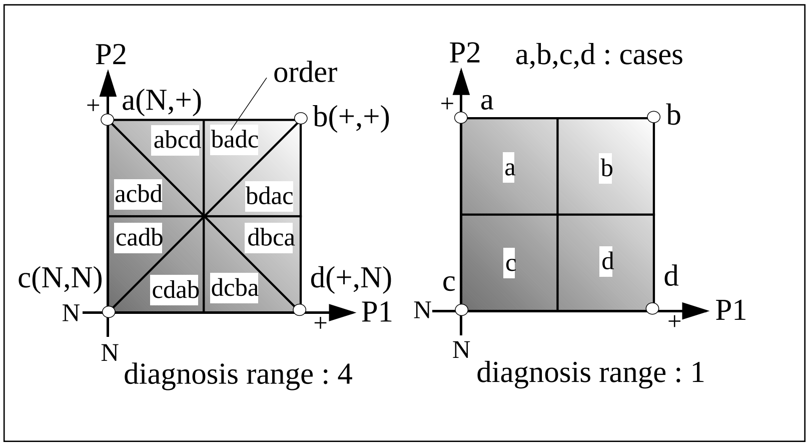


Figure 9. Example of Qualitative Space Division.
A. Diagnosis range 4. B. Diagnosis range 1.

ble. This order, *bd*, corresponds to an area that is a division of this quantity space. We can consider a certain number of cases in this ordering. This number is called the *diagnosis range* and, thus, determines the number of the division of the space.

For example, figures 9a and 9b show the spaces for which the 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.

$$N\max = 2^n \times {}_nP_2 \times \prod_{r=2}^{n-1}((n-r)+\sum_{k=2}^{r} {}_rC_k) \qquad (n \geq 3) \;\; (1)$$

Table 2 shows the result of the area division of a quantity space consisting of three sensors; the space is divided into 96 areas using the maximum diagnosis range of 8. 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 the control of an actuator means a transition of areas in the

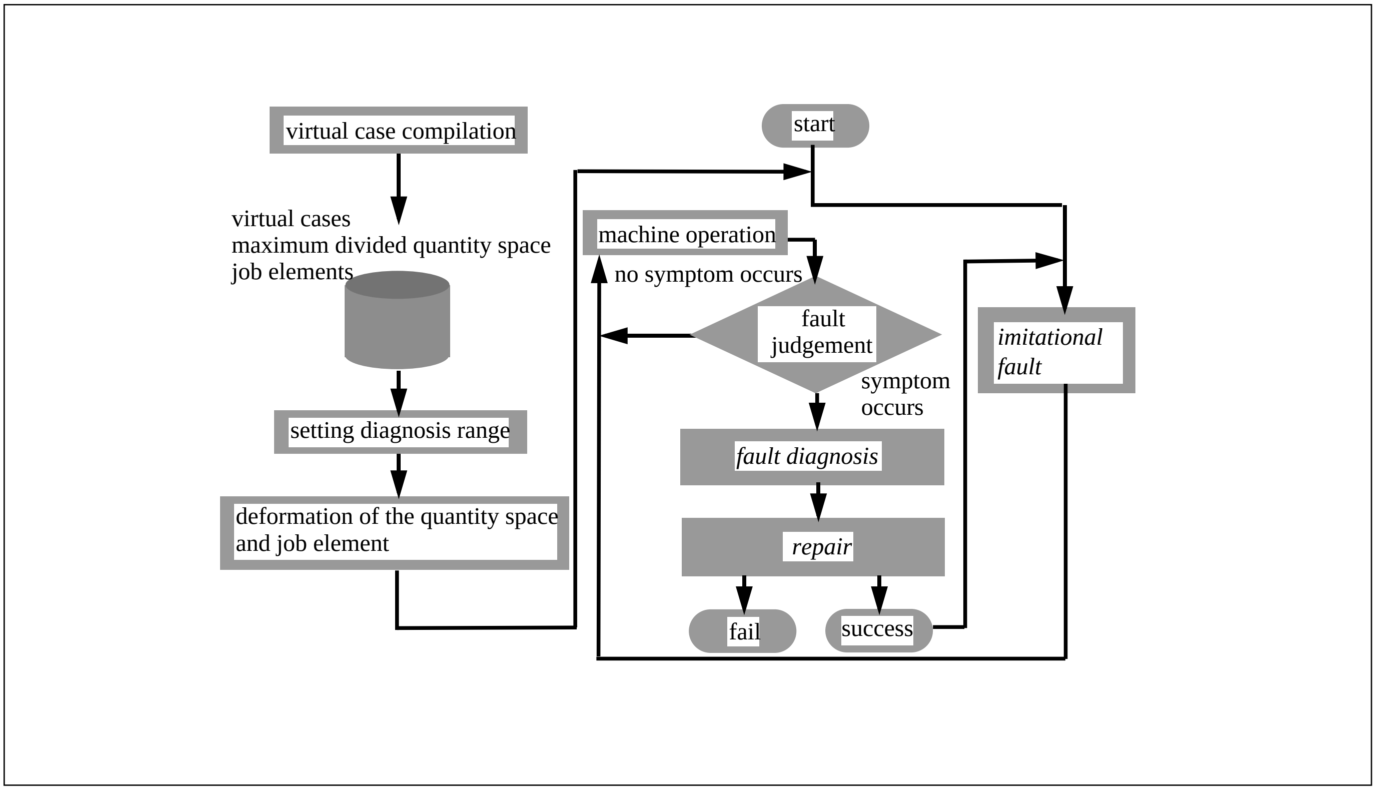


Figure 10. The Reasoning Algorithm of the Commercial Photocopier.

divided quantity space. However, there are possible and impossible 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 subsection, we explain the reasoning algorithm of the commercial photocopier. Figure 10 presents the reasoning algorithm.

Models and Knowledge During the development process, the following knowledge for each kind of machine must first be prepared: (1) the parameter model set up manually and the virtual cases compiled from the parameter model with the virtual case compiler and (2) the maximum divided quantity space and job elements.

Then, the following data must be determined: (1) the diagnosis range, (2) the divided quantity space and the job elements modified with respect to the diagnosis range, and (3) the fuzzy qualitative space map for each sensory parameter obtained through the imitational fault method.

Diagnosis The commercial version of the 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.

$$C = 1 - \sqrt{C(p_1)^2 + C(p_2)^2 + \ldots + C(p_n)^2} / \sqrt{n} \tag{2}$$

$$C(p_i) = Gm(q_i) - Gs(q_i) \qquad (i = 1, 2, \ldots, n) \tag{3}$$

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 follow-

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
↓	↓		↓			↓		↓

Table 4. Example of Fault Diagnosis.

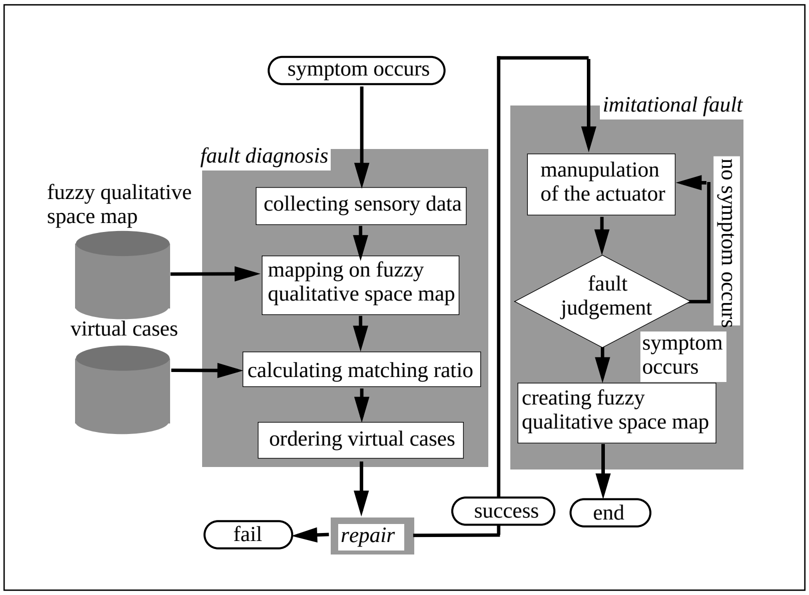


Figure 11. The Algorithm of Fault Diagnosis.

ing fuzzy qualitative data when the symptom "increase of parameter *Os*" occurs. In these data, the value of parameter *Ds*, for example, 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).

$$Ds(+ : 0.6, N : 0.4) \qquad (4)$$

$$Vs(+ : 0.6, N : 0.4) \qquad (5)$$

$$X(–0 : 8, N : 0.2) \qquad (6)$$

Thus, fault diagnosis 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:

First, the system searches for a path from the current area to the goal area by combining job elements. It is similar to techniques that have been developed in software analysis to reason out all execution paths that some code might take (Srimani and Malaiya 1992).

Second, it manipulates the actuators 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.

Third, 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, for example, because of the deterioration of the 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 developed a commercial photocopier based on the techniques described. Figures 14 and 15 depict its architecture and exterior. Figure 16 is a screen image of the virtual case

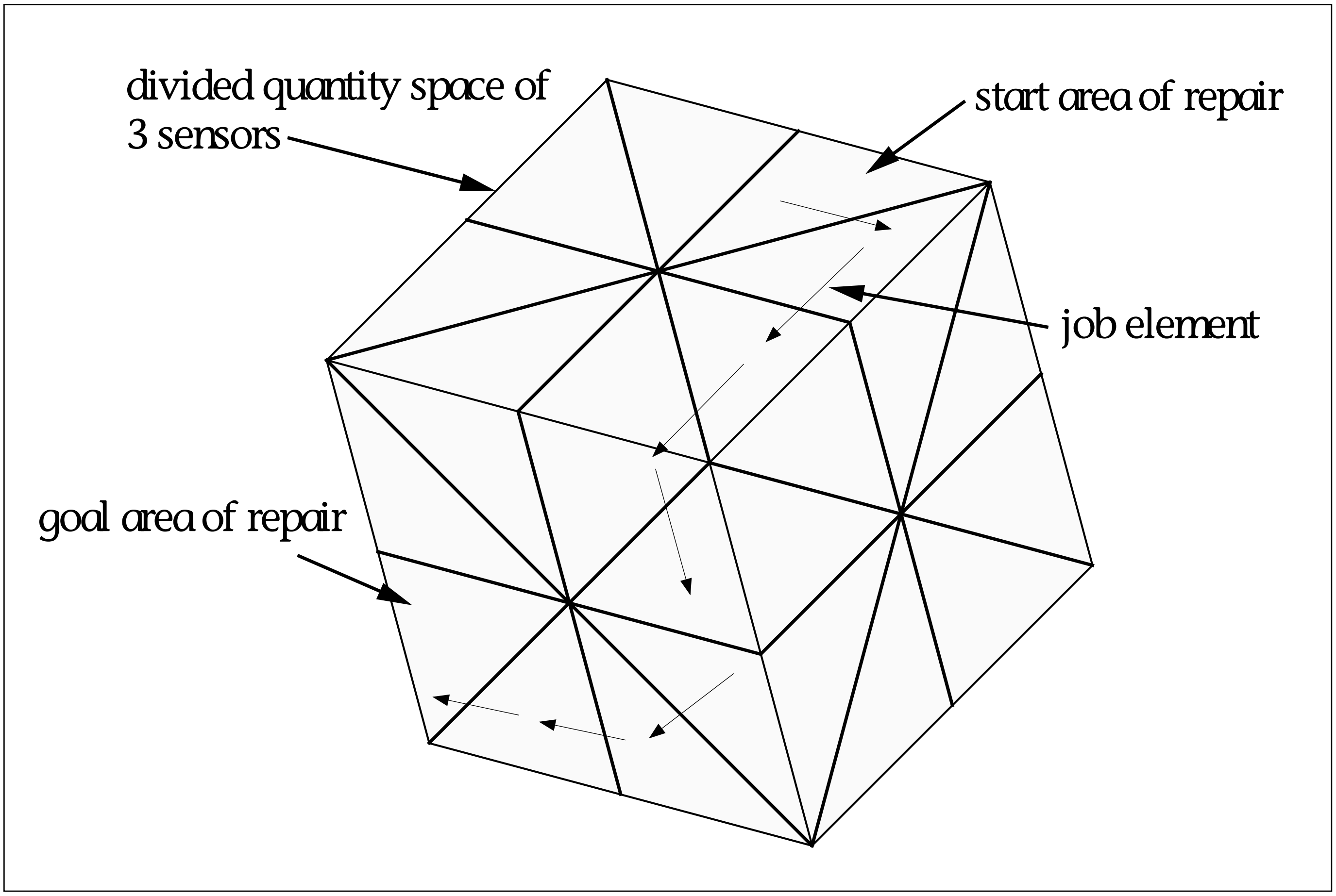


Figure 12. The Concept of Repair with Job Elements.

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 in the C and assembler languages and takes, at most, 10 minutes for a diagnosis and a repair, including a revision of the fuzzy qualitative space mapping. The system size is about 280 kilobytes, including 16 virtual cases; its diagnosis range is 2. These results are acceptable for an embedded system. Figure 18 shows typical results of some experimental diagnoses that demonstrate the diagnosis versus the fuzziness of the sensory information.

Although the knowledge-compilation techniques based on the virtual case greatly improve the reasoning speed and the system size, the system has less flexibility against faults with structural changes than simulation-based systems have.

The commercial photocopier has been available worldwide since April 1994. Although it is too early to evaluate the payoff of this product, we predict the decreases in downtime and maintenance costs paying for

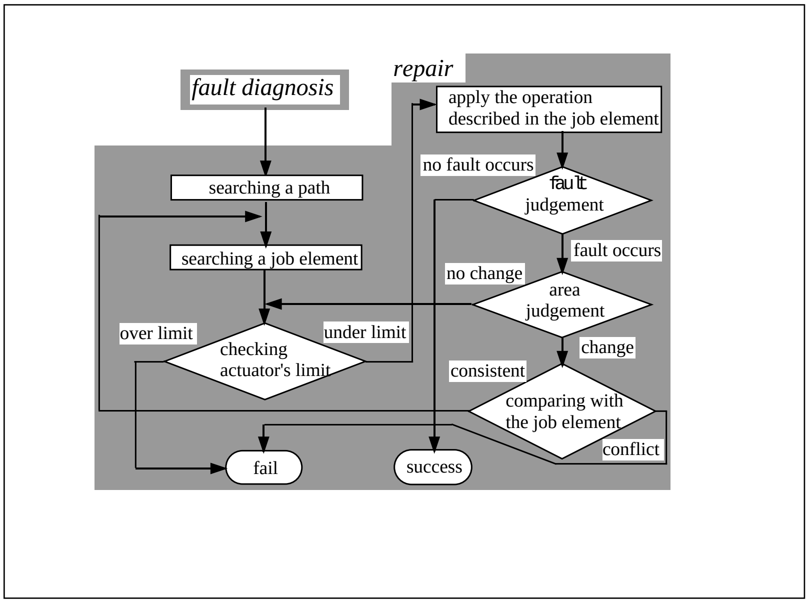


Figure 13. The Algorithm of the Fault Repairing.

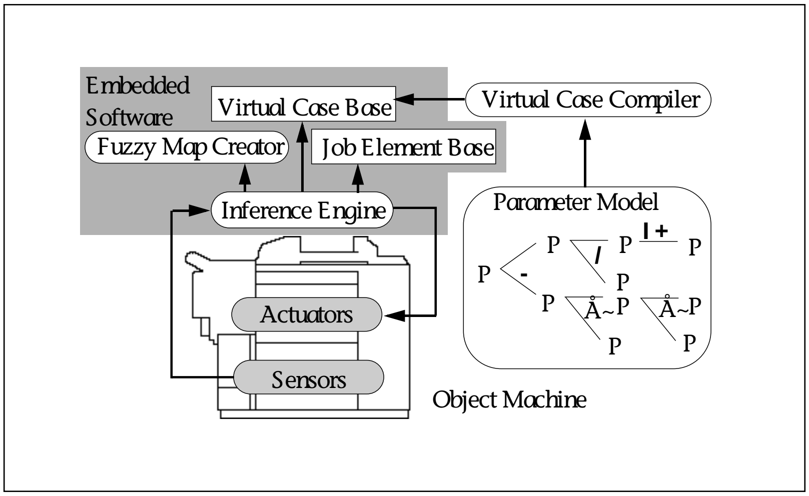
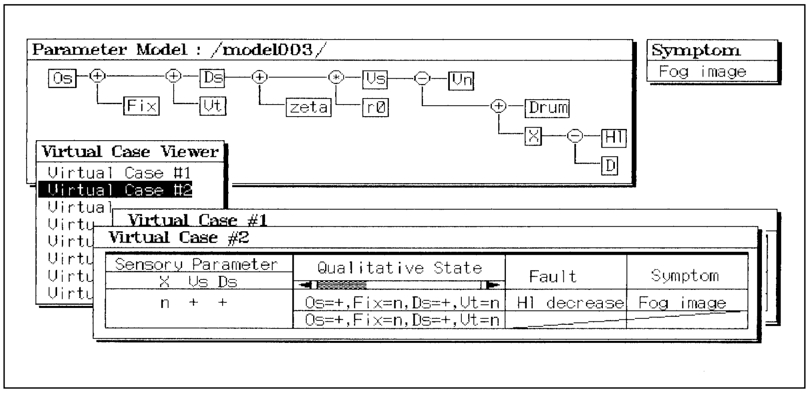


Figure 14. The Structure of the Commercial System.

Figure 15. The Exterior of the Commercial Machine.



Parameter Model : /model003/

Symptom
Fog image

Virtual Case Viewer
Virtual Case #1
Virtual Case #2
Virtual
Virtu
Virtu
Virtu
Virtu
Virtu

Virtual Case #1

Virtual Case #2

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

Figure 16. A Screen Image of the Virtual Case Compiler.

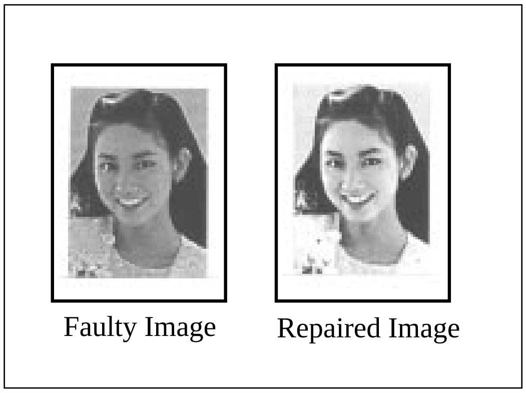


Figure 17. The Result of Fault Repairing.

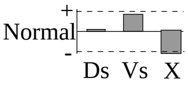
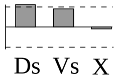
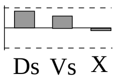
Quantitative Sensory Data				
Qualitative Value	Fuzzy	(+0.1,+0.9,-1.0)	(+1.0,+0.9,-0.1)	(+0.8,+0.6,-0.2)
	Non Fuzzy	(N,N,-)	(+,N,N)	(N,N,N)
Diagnosis Results	Fuzzy	○ halogen lamp	○ main charge	○ main charge
	Non Fuzzy	× transfer	× developing bias	× transfer

Figure 18. The Result of Fault Diagnosis.

the cost of development. Service engineers, particularly in the United States, prefer this self-maintenance photocopier to a traditional one because it can reduce their work and maintain the level of service.

Application Development, Deployment, and Maintenance

The commercial system was developed by 10 designers over 2 years following the development of the prototype system in the SMM Project at the University of Tokyo in 1992. The design team included mechanical, electric, and software engineers. The designers spent most of the development time on experiments for the sensors and software development.

We have not yet found any need for maintenance of the knowledge or the reasoning algorithm of the system.

Conclusion

In this article, we proposed a design strategy for the SMM and its prototype. We then proposed a method to reduce the system size and improve the performance of the reasoning based on the fuzzy qualitative reasoning and the imitational fault method. We also proposed a method to execute repair operations more accurately. We demonstrated the feasibility of a commercial version of the self-maintenance photocopier. Future projects include making repair planning more flexible by allowing users to select repair goals and developing a machine that can treat faults with structural changes based on the concept of functional redundancy (Umeda et al. 1994).

References

Bell, D.; Bobrow, D.; Falkenhainer, B.; Fromherz, M.; Saraswat, V.; Shirley, M. 1994. RAPPER: The Copier Modeling Project. In Proceedings of the Eighth International Workshop on Qualitative Reasoning about Physical Systems, ed. T. Nishida, 1–12. Unpublished manuscript.

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. Presented at the Thirteenth International Joint Conference on Artificial Intelligence Workshop on Engineering Problems for Qualitative Reasoning, Chambery, Savoie, France, 28 August to 2 September.

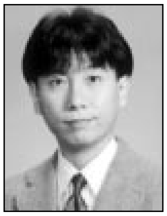
Srimani, P., and Malaiya, Y. 1992. Steps to Practical Reliability Measurement. *IEEE Software* 9(4): 10–12.

Umeda, Y.; Tomiyama, T.; and Yoshikawa, K. 1992. A Design Methodology for a Self-Maintenance Machine. In Proceedings of the First International Conference on Intelligent Systems Engineering, 179–184. London: The Institute of Electrical Engineers (IEE).

Umeda, Y.; Shimomura, Y.; Tomiyama, T.; and Yoshikawa, H. 1994. Using Functional Maintenance to Improve Fault Tolerance. *IEEE Expert* 2(3): 25–31.

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

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



Yoshiki Shimomura is a researcher at Mita Industrial. His research interests include functional reasoning and representation and intelligent mechanical systems. He has participated in the Self-Maintenance Machine Project at the University of Tokyo since 1988 and developed self-maintenance copiers as products of Mita Industrial. He graduated from the Kyusyu Institute of Technology in 1984.

Sadao Tanigawa is an engineer at Mita Industrial. He has participated in the Self-Maintenance Photocopier Project at Mita Industrial since 1990. He graduated from Oita University in 1988.

Yasushi Umeda is a lecturer at the Inverse Manufacturing Laboratory, Faculty of Engineering, the University of Tokyo. He received his doctor's degree in precision machinery engineering from the Graduate School of the University of Tokyo in 1992. His research interests include soft machines, green life-cycle design, intelligent computer-aided design for mechanical design, and functional reasoning.

Tetsuo Tomiyama has been an associate professor in the Department of Precision Machinery Engineering, the University of Tokyo, since 1987. From 1985 to 1987, he worked at the Centre for Mathematics and Computer Science in Amsterdam, The Netherlands. He received his doctor's degree in precision machinery engineering from the Graduate School of the University of Tokyo in 1985.