Agents to Guide Operators with Recognition of Time Series

Hiromitsu Shimakawa
Industrial Electronics & Systems Lab.
Mitsubishi Electric Corp.
Amagasaki, Hyogo, Japan

Yuzuru Sugitani and Kouji Kikkawa
Power & Industrial Systems Center
Mitsubishi Electric Corp.
Kobe, Hyogo, Japan

Taketoshi Watanabe
Mizushima Works
Kawasaki Steel Corp.
Kurashiki, Okayama, Japan

Abstract
A practical distributed AI system is proposed in this paper for the guidance of plant operations in a steel galvanizing plant. The system includes three autonomous agents for the guidance. One agent searches a plant operation to regulate the coating weight with fast plant variables. The others work to compensate faster plant variables with slower plant variables. Each of these agents views a monitored entity in the plant from an angle relevant to its tasks. It records a state of the entity in a new data structure explained in this paper. It examines the data structure to recognize trends. The experiment in a real plant has proved that the developed system can guide plant operations in much the same way as an experienced operator.

1. Introduction
A lot of effort has been made to put expert systems for plants into practice[8]. We have developed a practical distributed AI system[6]. Our target problem is guidance of plant operations in a steel-galvanizing plant where a steel plate is coated electrically. In the steel-galvanizing plant, coating weight is regulated with several decades of plant variables. The modification of one plant variable brings its effect in several seconds, while the modification of another in several hours. We refer to the former as a fast plant variable, and the latter as a slow plant variable. Fast plant variables are often modified in an emergency, because its immediacy is effective. The modification of fast plant variables, however, often prevents the preceding modification of a slow plant variable from producing an expected effect. It is very difficult to predict an effect of modifying slow plant variables. Thus, a proper mathematical model taking all plant variables into consideration has not been established in the steel galvanizing plant, though most of plants are well modeled for regulation.

In spite of lacking a proper mathematical model, experienced operators can regulate the coating weight manually. They distinguish the plant variables by whether they bring their effects immediately or not. The experienced operators use a fast plant variable for the regulation of the coating weight, while a slow plant variable for compensation of fast ones. From their experiences, they can select the appropriate moment at which fast plant variables would be compensated.

We think that the experienced operators execute several tasks simultaneously. We have developed a plant operation supporting system in which the tasks of the operators are accomplished by multi agents[4]. An agent applies rough mathematical models to fast plant variables to regulate the coating weight. Other agents propose to modify slow plant variables to compensate the fast plant variables. To select the appropriate moment to modify the plant variables properly, the compensation agents have to monitor entities in the plant, being conscious of the course of time.

In this paper, we propose a framework which can represent the course of time. We present agents to support plant operations. Each agent records data sampled every moment in an appropriate form for a task assigned to it. The data correspond to states of a monitored entity. Each agent recognizes not only a current state but also a state transition in the course of time to select the best moment for a plant operation. An experiment in a real plant has proved that the developed plant-operation-supporting system can guide plant operations in much the same way as an experienced operator.

2. Expertise in Steel Galvanizing Plant
2.1. Why AI System?
In the steel galvanizing plant, coating weight is regulated with several decades of plant variables. Some of them are shown in table 1. For each plant variable, the table indicates both a time-lag of effect appearance in the feedback value from modification of the reference value and a time-lag of effect appearance in the coating weight from the effect appearance in the feedback value. We refer to the time of the effect appearance in the coating weight from the
modifying the reference value of a plant variable as a response time of the plant variable. As the table illustrates, the response time varies from several seconds to several hours. A plant variable whose response time is short is a fast plant variable, while a plant variable whose response time is long is a slow plant variable.

In plant operations using the plant variables of various response times, it is very difficult to modify a slow plant variable. Operators have to wait for a long time until modification of a slow plant variable brings an effect. A fast plant variable is usually modified many times during the waiting time. Thus, the modification of the slow plant variable does not always bring an effect as it was expected. Therefore, the steel galvanizing plant is operated manually without a proper mathematical model on which all plant variables are modified. Actually, the plant works depending on experienced operators who can properly use several rough mathematical models and rules acquired from their experiences according to specific conditions.

The experienced operators have acquired strategies on which they properly use plant variables of various response times. We represent the strategies as heuristic rules.

### 2.2. Coating Control and Its Compensation

Many interviews with experienced operators have proved that they operate the plant on the following strategies.

1. They classify plant variables into three groups by their response time. The first is a group of short response times (several seconds). The second is a group of intermediate response times (several minutes), and the last is a group of long response time (several hours). We refer to plant variables of each group as fast plant variables, moderate plant variables, and slow plant variables, respectively.

2. They regulate the coating weight with the fast plant variables.

3. They compensate the fast plant variables with the moderate plant variables, and the moderate plant variables with the slow plant variables.

The plant operations described in (3) is needed, since every plant variable can take its value only in a specific range. For example, operators try to raise the present value of a fast plant variable, when it is effective in keeping the coating weight in a desirable range. However, it cannot be raised if it is close to the upper limit. An experienced operator would have decreased the value at an appropriate opportunity. The experienced operator would have modified a plant variable which has a longer response time, expecting that the modification would lead the plant to a situation where the value close to the upper limit can be decreased. He would have selected the appropriate moment for the compensation, examining whether an expected effect of the modification appears in a specified time.

We have developed a plant operation supporting system in which three agents act concurrently [see fig.1]. CoatingControl is an agent which searches a plant operation to regulate the coating weight with a short plant variable such as the current density and the line speed. PrimaryCompensation is an agent to compensate the fast plant variables with moderate plant variables such as the temperature and pH. SecondaryCompensation is an agent to compensate the moderate plant variables with slow plant variables such as the concentration of the solution for galvanization.

In fig.1, plant variables which enter into an agent from its left side and exit to its right side are controlled variables regulated with plant operations proposed by the agent. Plant variables which enter into an agent from its upper side are manipulated variables whose modification the agent proposes. Operators modify the reference value of the manipulated variable, expecting the modification would lead the feedback value of a controlled variable to a desirable range.

![fig.1 Agents for Regulation and Compensation](image)

### 3. Framework Conscious of Time

We have found that experienced operators monitor not only the present state of an entity in the plant, but also state transitions to the present time. We have developed a framework to represent a state transition of a monitored entity in the course of time.
3.1. Representation of State Transition

3.1.1. Instance and Concept: A set of variables whose values represent signals on a monitored entity defines a state of the entity. We refer to the set as an instance, and the variables as characteristic variables.

A concept is specified to declare types, names, and restrictions of the characteristic variables the instance is composed of. The restriction is the range to which the value of the characteristic variable belongs. The violation of the restriction means that the value of the characteristic variable has not been sampled.

An instance associated with concept tank shown in program.1 can represent a state of a monitored tank at every sample time-point. A state of the tank is represented by the feedback value of the coating weight and the line speed, the upper limit of the line speed, the lower limit of the line speed, the feedback value of the temperature, the difference of the reference value of the temperature from the previous one.

The range associated with each characteristic variable in program.1 represents its restriction. In this example, the tank has variable limits on the line speed. Since restrictions cannot represent variable limits, extra characteristic variables (LS_upLim, LS_lowLim) are declared. The relationships between these limits and the line speed are represented by another kind of condition[see 3.2].

```cpp
concept tank {
    int coat: >= 0, <= 1000;
    int LS: >= 0, <= 300;
    int LS_upLim: >= 0, <= 300;
    int LS_lowLim: >= 0, <= 300;
    int temp: >= 0, <= 100;
    int temp_chg: >= -100, <= 100;
}
```

3.1.2. Scene and Series: A monitored entity lies in various states in the course of time. The course of time is represented by a time-axis. A period is an interval in the time-axis[7]. The period p is a pair of the starting terminal point ts and the ending terminal point te, which are represented by relative positions from a specific basis in the time-axis, that is

\[ p = (ts, te) \]

A scene s of a specific concept is a pair of instance I of the concept and a period p,

\[ s = (I, p) \]

where the instance remains in one state.

A series is proposed to represent the state transition of the monitored entity in a certain period. A series \( \sigma \) is a chronologically ordered set of scenes of the concept.

\[ \sigma = (s_1, s_2, ..., s_n) \]

The length of the series is the number of the scenes in it. A series has a specific basis in the time-axis. It is the basis of starting terminal points and ending terminal points of all scenes in the series as shown in fig.2.

Suppose that t is any of starting terminal points or ending terminal points of scenes in a series. We assume that t is represented by \( b + n \cdot c \) if t satisfies

\[ b + n \cdot c \leq t < b + (n + 1) \cdot c, \]

where b is the basis of the series, n is non negative integer, and c is a positive constant[see fig.2]. The assumption synchronizes all sample time-points. We refer to c as a clock of the series. The clock determines the accuracy with which the series represents the state transition of the monitored entity.

When signals are sampled, a scene recording the signals is coupled to the back of a series. Suppose that signals are sampled at sample time-point t such that \( t = b + (n + 1) \cdot c \), when s is the last scene in a series and its period p is represented by \( (b + mc, b + (n + 1)c) \). If the instance stays in a same state in p and at t, a scene whose period is represented by \( (b + mc, b + (n + 1)c) \) is recorded instead of s.

3.2. skeletons

We refer to a characteristic of state transitions as a trend, experienced operators make their decisions based on trends. They distinguish series which satisfy specific conditions from others.

We provide two kinds of conditions. As mentioned above, the restriction is the range to which the value of the characteristic variable should belong. A constraint is a lin-
ear inequality more than one characteristic variables should satisfy. For example, restrictions are conditions following "such that" and a constraint is a condition following "under" in program.2. The constraint represents the relationship between the line speed and its limits.

Using restrictions and constraints, a concept skeleton defines a condition on instances of a specific concept. It corresponds to all instances which satisfy the condition. Concept skeleton "reducibleLineSpeed" in program.2 defines the condition required for the line speed to be reduced.

A series skeleton toBeRepaired tank in program.2. Skeletons

A series skeleton defines a trend. A series skeleton toBeRepaired tank in program.2. Skeletons indicates state transitions of the tank after increment of a reference value of temperature. It specifies that the line speed should be reduced because the preceding plant operation brings an expected effect; after the increment of the temperature, the tank stays in states associated with "normal" for about 30 seconds, and in states associated with "reducibleLineSpeed" for about 15 seconds. An appropriate moment to properly modify the line speed is selected using this series skeleton. A sequence of periods is convenient to recognize a time required in a chemical reaction.

3.3. Matching

A series skeleton represents a trend. The unification of a series with a series skeleton allows us to recognize a trend[9]. The unification consists of two kinds of matching.

Scene matching is to check which concept skeletons specified in a series skeleton are satisfied by an instance state recorded in each scene. The scene matching clarifies a period where each concept skeleton are satisfied. The time-axis matching is to check whether the clarified period meets the periods specified in the series skeleton.

4. An Agent with Series: Specialist Module

4.1. Facilities of Specialist Modules

We introduce specialist modules as agents to support plant operations. General features of agents based on distributed AI methods are explained in [6]. The features include:

- **Concurrency** A target problem is solved with concurrent execution of many tasks. Several agents act simultaneously for the execution. Each agent is autonomous.

- **Modularity** Activities of agents are encapsulated in object-oriented manners. Agents have procedures and local memories hidden from outside. They communicate with each other using messages[5]. The interaction between them are limited to the message passing.

- **Specialization** Agents have heuristic rules to carry out the tasks assigned to them. They would accomplish only tasks relevant to them, and neglect others. The target problem is solved with cooperation of them.

Specialist modules are based on agents that are proposed in MACE[4], but they have functions specific to the recognition of series. Each specialist module can keep series of a specific concept in its local memory. When it receives a message whose content is a newly sampled scene, it couples the scene to the back of the internal series. If the series too long to be held in the specialist module, an early part of the series is stored in the external file, and an entry of the file is registered in the specialist module. The entry is a key to retrieve the early part of the series. Each specialist module contains procedures called as methods which are activated by either external messages or events such as timer alarms. It can set a timer to activate a method for a regular recognition of its internal series. In this method, it compares the series with several series skeletons to examine whether the series has any trends. It selects a plant operation according to the recognition.

The whole task for supporting plant operations is dis-
tributed to several specialist modules. An operation proposed by a specialist module should not conflict with operations proposed by others. Specialist modules pass messages to solve conflicts as described in 4.3.

4.2. View of Specialist

A plant operation supporting system should consist of efficient components so that it may work as a real-time system. Each specialist module should record the state transition of the monitored entity in an efficient way to carry out a task assigned to it.

For example, experienced operators often make their decisions by examining only whether the temperature is increased or decreased by more than 1.0°C. What should be recorded is one of "increment", "decrement", and "no operation". Experienced operators would examine the entity state not at the figure level but at the category level. Categories of states can be represented by concept skeletons.

We think it important that a series is recognized by two stages. The scene matching stage determines concept skeletons an instance matches with. The record of indices of matched concept skeletons would make time-axis matching possible. Let us consider a new concept which declares characteristic variables representing the matched concept skeletons. This concept is acquired when the original concept is viewed from an angle relevant to the task assigned to the specialist module. We refer to the concept as an aspect of the original concept.

An aspect strongly depends on a task. Each specialist module increases its efficiency, viewing the state transition of the monitored entity in a way specialized to its task. An aspect is the reduction of an original concept to a part relevant to the task. In the actual observation, the instance state seldom transit categories defined by concept skeletons. The aspect would greatly compress the length of series (see 3.1.2).

4.3. Management of Future Series

Several specialist modules independently selects a plant operation. A plant operation of one specialist module may conflict with that of another specialist module. A conflict means more than one plant operations on one plant variable in a specific future period. An arbitration mechanism is needed so that each specialist module may not conflict with others in plant operations. Conflicts must be solved in all future periods. Dechter proposed a temporal constraint network in [2] to maintain the truth in the temporal relationships. We use a series for this purpose, because the mechanism to record a past state transition can be reused.

Each specialist module presents values of plant variables in a future period as its proposing plant operation. It can be regarded as a scene in the future. Let us think of an arbitrator which is a kind of a specialist module containing a series to represent future state transition of a monitored entity. It manages the future series as a time table of the plant variables.

When a specialist module for guidance finds a plant operation, it sends a message to the arbitrator. The message is a requirement for registering a plant variable value in an associated future period. The arbitrator tries to register the plant variable value in the designated period. If the period is not already occupied by any other value, the register succeeds. Otherwise, the register fails, which means occurrence of a conflict.

Priorities has to be taken into consideration to solve a conflict. However, a method to determine a suitable priorities strongly depends on each problem. We have not find a general procedure to determine priorities, yet. The simplest method to solve a conflict is to make the specialist module trying the last register abandon its plant operation.

5. Practice in Real Plant

5.1. System

We have developed a plant operation supporting system consisting of the components shown in fig.3 on a SONY NEWS 1750(NewsOS 3.3). The message-passing mechanism is implemented on RPC[1].

The numbers of the characteristic variables of aspects on which CoatingControl, PrimaryCompensation, and SecondaryCompensation search plant operations are 6, 9, and
8, respectively. A cycle of the regular recognition of the internal series in each specialist module depends on the response time of the plant variables the specialist module treat as manipulated variables. CoatingControl, PrimaryCompensation, and SecondaryCompensation have to activate their methods every 10 seconds, every 30 seconds, and every 600 seconds, respectively.

Data Sampler (DS) transfers some of 26 kinds of plant variable values sampled every 3 seconds with a time-stamp to DM, CoatingControl, PrimaryCompensation, and SecondaryCompensation. In Data Manager (DM), all plant variable values transferred from DS are stored in a series without conversion, while the three specialist modules convert the values into scenes of their own aspect. Graph tools, which are constructed on X window system, require DM for plant variable values in a specified period. Future Resource Manager (FRM) is a specialist module organizing plant operations proposed by the three specialist modules for guidances.

5.2. Knowledge

Each of the specialist modules to propose plant operations has knowledge which is useful for accomplishing an assigned task. The knowledge is represented by a set of rules in each specialist module. The condition part of a rule is represented by series skeletons. The guidance message in the action part is presented to an operator.

5.2.1. Mathematical Model: CoatingControl has several rough mathematical models which are applicable to some of fast plant variables according to specific conditions. It assumes that slow and moderate plant variables would have little effect on the coating weight, until an effect of modifying a fast plant variable appears. One of the models of CoatingControl is explained briefly.

Assume that the consumed rate $r$ (mol/s) of the coating metal $M$ depends on two kinds of fast plant variables: the current density $e$ (A/m$^2$) and the line speed $L$ (m/s) [see fig.4]. The average coating weight per a unit area $d$ (mg/m$^2$) shown in fig.5 is supposed to be proportionate to $e$; that is,

$$d = \varepsilon \cdot e,$$

where $\varepsilon$ is a constant. $M$ is included in the whole coating substance by the proportion $n$ ($0 \leq n \leq 1$). Using the width of the steel plate $w$ (m), $r$ is represented by

$$r = \varepsilon \cdot e \cdot (w \cdot L) \cdot n. \quad (1)$$

The concentration of the solution entering into the tank is $C_i$, while the concentration of the solution in the tank is $C$. In the tank, $M$ increases by $s \cdot C_i$ and decrease by $s \cdot C + r$. Therefore,

$$V \cdot \frac{dC}{dt} = s \cdot C_i - s \cdot C - r,$$

where $V$ (m$^3$) is the volume of the solution and $s$ (m$^3$/s) is the flow speed in the tank. Since $e$ and $L$ is far faster than $C$, $C$ is regarded as constant while $e$ and $L$ are changing. Thus,

$$r = s \cdot C_i - s \cdot C. \quad (2)$$

From (1) and (2), a pair of $e$ and $L$ is computable because other variables are measurable.

In reality, CoatingControl contains a series of an aspect whose characteristic variable is computed from all fast plant variable values. CoatingControl regularly examines the series to determine whether modification is needed. Upon modification, CoatingControl selects a proper model with rules, and proposes a value of the plant variable calculated from the model.

5.2.2. Compensation: When the feedback value of a fast plant variable is close to its limit, PrimaryCompensation searches a plant operation compensating it with a moderate plant variable. The specialist module expects a situation where the fast plant variable can be modified without undesirable effects. When an expected situation actually appears, it modifies the fast plant variable away from its limit. In same ways, SecondaryCompensation compensates moderate plant variables with slow plant variables.

It is difficult to establish a proper mathematical model for compensation, because it modifies plant variables whose effects do not appear immediately. PrimaryCompensation and SecondaryCompensation have heuristic rules based on the operator's experiences. They propose plant operations only when specific conditions are satisfied.

For example, SecondaryCompensation has the rules shown in program.3. Plant variables are generally compensated with a sequence of planned plant operations. The early parts of the charts shown in fig.6 illustrates a situation in which Rule1 is applicable. The coating weight is
Rule1:
if((the feedback value of coating weight has been within the lower part of the normal range for a while) && (the feedback value of line speed is normal) && (the feedback value of current density is normal) && (the feedback value of temperature is higher than a normal upper limit))
then take a plant operation:
(raise the reference value of concentration).

Rule2:
if((after an hour from increment of concentration, a feedback value of coating weight is within the upper part of a normal range) && (the feedback value of line speed is normal) && (the feedback value of current density is normal) && (the feedback value of temperature is higher than a normal upper limit))
then take a plant operation:
(pull the reference value of temperature down).

Examples of Rules
To be easily understood, the periods and plant variables are represented by their meaning.

The number of rules in CoatingControl, PrimaryCompensation, and SecondaryCompensation is 11, 27, and 19, respectively. The rules are very few because of two features of this system. One is the specification of rules with series skeletons. The series skeletons enable the rules to concisely specify conditions for selection of moments to modify plant variables. The other is the classification of plant variables by their response time, accompanied with assignment of a clear role to every group of plant variables. The regulation of the coating weight is the role of the fast group, and the compensation is the role of the other groups. The second feature encapsulates rules for each group into one specialist module. Thus, the rules in one specialist module can be described with a little consideration of the rules in other specialist modules.

5.3. Coordination
Plant operations proposed by the specialist modules for guidances might sometimes conflict. They are coordinated by an arbitrator:FRM. FRM manages a future series to find a conflict. It uses a simple method to solve a conflict: if a conflict occurs, a specialist module which sends a message requiring the last register abandons its plant operation. When FRM finds a conflict, it returns an answer, which causes the sender specialist module to search for another plant operation. The sender specialist module resumes searching a plant operation. It usually has several substitutes. When it cannot find any substitutes, it commits the judgment to an operator.

5.4. Response
We compress a series in each specialist module with an aspect. Since this would diminish the number of access times of a specialist module to the external files, we can

![fig.6 Application of Rules](image)

![fig.7 Response Time of Specialist Module](image)
expect a decrease in the response time. We measured the response time of a specialist module to examine the effectiveness of the compression of series with an aspect.

We measured the response time of CoatingControl, because it has the simplest task of the three specialist modules. We prepared 2 versions of CoatingControl; one contains an uncompressed series and the other contains a compressed series with an aspect. We measured the response time of the both versions, having a plant simulator generate 26 kinds of signals every second. The plant simulator generated different data every 30 seconds. The length of a series each version held in it was 30, that is, each version held 30 scenes in it. The response time was measured in 6 kinds of series whose periods are 50 seconds, 100 second, 200 seconds, 500 seconds, 700 seconds and 1000 seconds. For each case, the response time was measured more than 3 times. We plot the worst response time for each case [see fig. 7]. The response time is greatly reduced by compression as expected.

5.5. Result
We have verified the practicability of the developed plant-operation-supporting system in a real plant. The plant was operated by an experienced operator who could not see the plant-operation-supporting system. At the same time, the system proposed plant operations for the plant. We logged both plant operations. Later, the log of the system was compared with that of the operator.

In this experiment, three specialist modules proposed 539 plant operations in all. 39 plant operations of them indicated to modify specific plant variables. 7 plant operations of them were not identical with what the operator actually did. To the contrary, the experienced operator modified some plant variables 4 times, while the plant operation supporting system did not propose the modification of any plant variable.

The percent of the proper modification proposed by the plant operation supporting system is the following.

\[
\frac{(39 - 7)}{39} \times 100 = 82.05\%.
\]

The percent of the proper plant operations proposed by the plant operation supporting system was the following.

\[
\frac{(539 - (7 + 4))}{539} \times 100 = 97.96\%.
\]

These figures prove that the plant-operation-supporting system is useful.

6 of the plant operations which were not identical with what the operator actually did were proposed when a welded part of steel plates was passing in the tank. Different strategies are needed to regulate the coating weight in such a situation. Switching between strategies depending on situations is our future work.

6. Conclusion
We have developed a plant-operation-supporting system, using a distributed AI method. We take notice of expertise an experienced operator has for the steel galvanizing plant. They regulate the coating weight with fast plant variables, while they use plant variables whose effects does not appear immediately for the compensation.

In this paper, we explained two kinds of agents for guidance. One agent proposes the modification of the fast plant variables to regulate the coating weight, based on rough mathematical models. The others propose how to compensate faster plant variables with slower plant variables, based on heuristic rules. Series are explained to represent state transition of the monitored entity in the plant. Matching of series with series skeletons are tried to examine trends. The agents contains series and try the matching. The aspect is introduced to accomplish a task assigned to each agent effectively. An agent managing a future series solves conflicts between plant operations proposed by the agents. The experiment in a real plant has proved that the developed plant operation supporting system is useful.

Switching between strategies depending on situations is our future work.

References