한국경영과학회/대한산업공학회 2003 춘계공동학술대회
2003년 5월 16일-17일 한동대학교(포항)

Goal-formation Process in Fractal Manufacturing Systems

Kwangyeol Ryu and Mooyoung Jung
Advanced Product and Production Technology Center
Division of Mechanical and Industrial Engineering
Pohang University of Science & Technology (POSTECH)

Abstract

Decomposition of tasks in the ordinary manufacturing systems is usually based on the predefined goal of the system. To achieve the high-level-goals (e.g., factory goal or company goal), several sub-goals should be achieved in advance. However, goals can change along with the current status of the system and the external environmental situations. Thus, a manufacturing system should support the goal-formations which can be bearable these changes for efficient and effective operations. Therefore, it is necessary to develop a systematic methodology for the goal-formations in a manufacturing system. Especially, the formation and/or change of goals in real-time should be possible for distributed and dynamic systems including the fractal manufacturing system (FrMS). In this paper, a threefold methodology is proposed for the goal-formation process (GFP) in the FrMS; 1) a goal-generating process (GGP) to make and propagate fuzzy goals, 2) a goal-harmonizing process (GHP) to eliminate or reduce conflicts and interferences of goals by using a mobile agent-based negotiation scheme, and 3) a goal-balancing process (GBP) to make a compromise between goals by using quantifiable indicators of the manufacturing system.

Keywords: Goal-formation process(GFP), Fractal manufacturing system(FrMS), fractal, negotiation

1. Introduction

Facing intensified competition in a growing global market, manufacturing enterprises have been reengineering their production systems to survive by developing more intelligent computer integrated manufacturing (CIM) systems. Major goals of CIM include, but are not necessarily limited to, lowering manufacturing costs, rapidly responding to changing customer demands, shortening lead times, and increasing the quality of products (Chang et al. [1998], Cho [1993], and Son [2000]). However, the development of a CIM system is an incredibly complex activity, and the evolution to CIM has been slower than expected (Mettala [1989] and Smith [1992]).

As a control model for implementing CIM systems, hierarchical decomposition of shop floor activities has been commonly used in the shop floor control system (SFCS). Hierarchical control is easy to understand and less redundant than other distributed control architectures such as heterarchical or hybrid control. However, it has a crucial drawback, which is that a small change in one level may significantly and adversely affect the other levels in the hierarchy. In other words, it means that it is difficult to meet dynamically changing customer requirements because the hierarchical control architecture is not flexible enough to handle dynamic reconfiguration of the shop. Therefore, it is normally said that a hierarchical control architecture is much more suitable for production in a steady environment than in a dynamically changing environment.

To cope with such difficulties, the future manufacturing system should be flexible, highly reconfigurable, and easily adaptable to the dynamically changing environment. Furthermore, it should be an intelligent, autonomous, and distributed system composed of independent functional modules. To meet these requirements, several manufacturing paradigms have been newly proposed including a bionic manufacturing system (BMS; Okino [1993] and Ueda [1992]), a holonic manufacturing system (HMS; Brussel et al. [1998] and Seidel et al. [1994]), and a fractal manufacturing system (FrMS; Ryu et al. [2000], Ryu et al. [2001], Tirpak et al. [1992], and Warnecke [1993]). Tharumarajah et al. [1996] provides a comprehensive comparison among a BMS, a HMS, and a FrMS in terms of design and operational features.

The FrMS is mainly concerned in this paper, because it has been known to have the most advantageous features; nevertheless, there is little researchers who have an interest on it. The FrMS is the flexible manufacturing system which is driven by goal-oriented manners. In order to facilitate goal-formation process (GFP) of the FrMS, several schemes have been proposed in this paper regarding a goal-generating process (GGP), goal-harmonizing process (GHP), and goal-balancing process (GBP) which are systematically designed and autonomously performed based on cooperation and negotiation between system components. After reviewing basic concept and characteristics of the FrMS in Section 2, goal-formation process (GFP) in the FrMS will be precisely described in Section 3 before concluding this paper.

2. Fractal Manufacturing Systems(FrMS)

An FrMS is a new manufacturing concept evolved from the *fractal factory* introduced by Warnecke [1993]. It is based on the concept of autonomously cooperating multi-agents referred to as *fractals*. The basic component of the FrMS, named as a basic fractal unit (BFU), consists of five functional components; an observer, an analyzer, a resolver, an organizer, and a reporter (Ryu et al. [2000] and Ryu et al. [2001]). The fractal architecture represents a hierarchical structure built from the elements of a BFU, and the design of a basic unit incorporates a set of pertinent attributes that can fully represent any level in the hierarchy (Tirpak et al. [1992]). In other words, the term 'fractal' can represent every bit of the system from an entire manufacturing shop at the highest level to a physical machine at the bottom level. Each BFU provides services according to an individual-level goal and acts independently while attempting to achieve the shop-level goal

Figure 1 depicts the overview of the FrMS. Every controller at every level in the system has a self-similar functional structure. After the initial setup of a system, the configuration of the system may need to be reorganized in response to environmental changes including dynamic change of customer orders or unexpected events such as machine breakdowns. In these cases, fractals autonomously and dynamically change their structure, via the actions of agents based on the appropriate working mechanisms. Figure 1 shows two facility layouts and the corresponding compositions of fractals before and after the restructuring process.

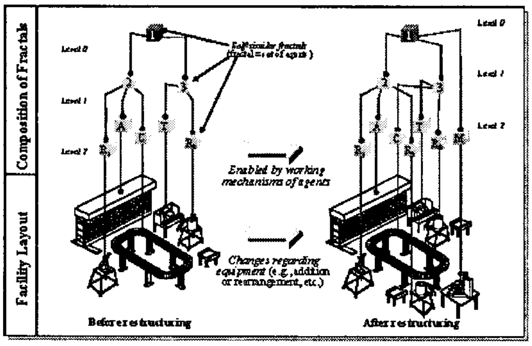

Figure 1. Reorganization of the system using a dynamic restructuring process in the FrMS

2.1 Definitions

The formal definition of a fractal was first mentioned by Warnecke[1993]. He defines a fractal as 'an independently acting corporate entity whose goal and performance can be precisely described'. This definition provides an opportunity to understand the general concept of the FrMS. However, the definition mentioned above is thought to be only concerned in the theoretical viewpoints of manufacturing systems. By considering implementation aspects of the system, definitions of a fractal and the FrMS can be refined as followings;

- A fractal is a set of self-similar agents whose goal can be achieved through cooperation, coordination, and negotiation with others, and it can reorganize the configuration of the fractal system to a more efficient and effective one.
- The FrMS is a flexible and fault-tolerant system developed and operated under the fractal architecture.

2.2 Basic Fractal Unit(BFU)

A fractal consists of five functional modules including an observer, an analyzer, a resolver, an organizer, and a reporter. Functional modules and their relationships in a fractal are illustrated in Figure 2. The functions of each module can be defined depending upon the application domain. However, when the target domain is determined, the main functions of each module will be consistent throughout the system. Figure 2 illustrates a bottom-level fractal whose functions are similar to those of a conventional equipment controller in a SFCS. The fractal directly connected to equipment (e.g., machine, robot, etc.) gets sensory signals from equipment and sends messages or commands to them.

The function of an *observer* is to monitor the state of the unit, to receive messages and information from outer fractals, and to transmit composite information to correspondent fractals.

The function of an *analyzer* is to analyze alternative job profiles with status information, to rate dispatching rules, and to simulate analyzed job profiles in real-time. The analyzer finally reports results to the resolver so that the resolver can use them to make decisions.

A *resolver* plays the most important role in a fractal, generating job profiles, goal-formation processes, and decision-making processes. During goal-formation processes, the resolver may employ a variety of numerical optimization or heuristic techniques to optimize the fractal's goal. If necessary, the resolver executes negotiations, cooperation, and coordination among fractals.

The function of an *organizer* is to manage status and addresses of fractals, particularly for dynamic restructuring processes. The organizer may adopt numerical optimization techniques to

Figure 2. Functional modules and relationships of a BFU in the FrMS

find the most profitable configuration for the shop floor while reconfiguring fractals. The fractal status is used to select the best job profile among several alternatives, and the fractal address is used to find the physical address of the fractal (e.g., *machine_name*, *port_number*) on the network.

Lastly, the function of a *reporter* is to report results from all processes in a fractal to others. In the case of a bottom-level controller, as illustrated in Figure 2, the fractal is similar to a traditional equipment controller. Therefore, most of messages are commands for controlling the hardware.

2.3 Characteristics of a fractal

Characteristics that differentiate an FrMS from other manufacturing systems include self-similarity, self-organization, goal-orientation, etc. Specifically, the dynamic restructuring process (DRP), which is a representative method for self-organization, and goal-formation process facilitating goal-orientation are the most distinctive characteristics among them.

2.3.1 Self-similarity

The characteristic of self-similarity refers not only to the structural characteristics of organizational design, but also circumscribes the manner of performing a job (service), as well as the formulation and pursuance of goals (Warnecke [1993]). To achieve goals in a manufacturing environment, there can be many possible solutions to individual problems. Even though there are components with the same goal in the system, conditions or situations of them may be different. This makes fractals with identical goals even though input and output variables may have quite different internal structures. Fractals, which have different internal structures, are 'self-similar' if they can make same outputs with the same inputs without considering their structures. Self-similarity should be regarded in view of a functional structure, not of the structure of the physical equipment.

2.3.2 Self-organization

Self-organization in the FrMS affects both the theoretical and operational methods. The theoretical method (self-optimization) means the application of suitable methods for controlling processes and optimizing the composition of fractals in the system. The operational method (dynamic restructuring process; DRP) supports the reconfiguration of network connections and the reorganization of fractals in the system. When the self-optimization module informs the necessity of reconfiguration, existing fractals (eF_A and eF_B in Figure 3) first change network connections based on the frequency of interactions, and reorganize their structure to make newly configured fractals (nF_A, nF_B, nF_C) with higher stability.

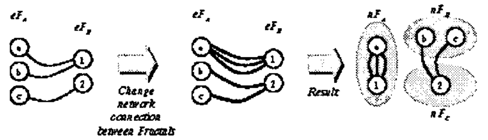

Figure 3. Dynamic restructuring process (DRP)

2.3.3 Goal-orientation

Every fractal in the FrMS has individual goals. The goals of each fractal are somewhat different from those of the others. To coherently achieve their goals, goal consistency should be maintained. The goal-formation process(GFP) is a process of generating goals by coordination processes between participating fractals and modifying them as needed. The GFP is supported by an inheritance mechanism to ensure goal consistency. Warnecke[1993] pointed out that the GFP is a reliable method for revealing any conflicts between competing goals. The FrMS must continue to autonomously develop goals in order to harmonize the system by resolving conflicts. Figure 4 shows the GFP in the FrMS. The goal of the FrMS can be achieved in an iterative fashion by developing individual goals of each fractal and by getting feedbacks after achieving each goal.

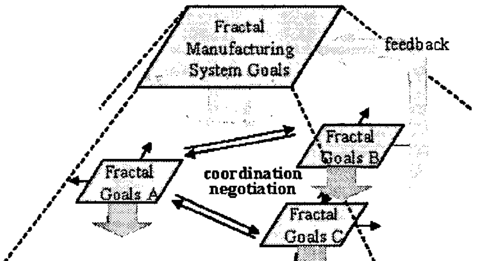

Figure 4. Goal-formation process (GFP) in FrMS

2.4 Agent-based FrMS

The FrMS continuously reorganizes the system configuration to be an optimal state with respect to the dynamically changing environment. To reconfigure the system, network connections between fractals can be flexibly reconnected. To consider the future implementation of the FrMS, an agent technology is adopted for facilitating easy change of network connections so that the system can autonomously improve its structure. With the agent technology, servers can be upgraded, services moved, load balancing interposed, and security policy enforced, without interruptions or revisions to the network and clients (Wong et al. [2001])

The types and functions of agents that implement functional modules of a FrMS have been published in the earlier literature (Ryu et al. [2001], Ryu and Jung [2002]). Totally, 18 agents have been defined and modelled by using Unified Modeling Language (UML). Refer to the published literature regarding the detailed functions of each agent (their brief functions can be intuitively inspired from their names). The list of agent names is as follows (note that "-M" and "-S" written after the abbreviated name of each agent represent mobile agents and software agents respectively);

- Agents for an Observer
  - Network Monitoring Agent (NMA-S)
  - Equipment Monitoring Agent (EMA-S)
- Agents for an Analyzer
  - Schedule Evaluation Agent (SEA-S)
  - Dispatching-rule Rating Agent (DRA-S)
  - Real-time Simulation Agent (RSA-S)
- Agents for a Resolver
  - Schedule Generation Agent (SGA-M)
  - Goal-Formation Agent (GFA-S)
  - Task Governing Agent (TGA-S)
  - NEgotiation Agent (NEA-M)
  - Knowledge Database Agent (KDA-M)
  - Decision-Making Agent (DMA-S)
- Agents for an Organizer
  - Fractal Status Manager (FSM-S)
  - Fractal Address Manager (FAM-S)
  - REstructuring Agent (REA-M)
- Agents for a Reporter
  - Network Command Agent (NCA-M)
  - Equipment Command Agent (ECA-S)
- Miscellaneous Agents
  - SysTem Agent (STA-S)
  - NeTwork Agent (NTA-S)

Figure 5 illustrates the class diagram of fractal agents considered in this research. To simplify the diagram, the attributes and operations of the classes are omitted in the figure. Several other agents supporting major fractal agents are also omitted to focus on the relationships between fractal agents.

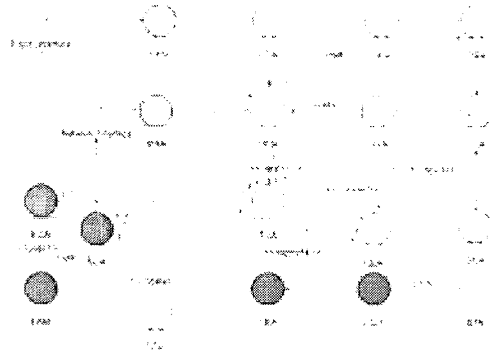

Figure 5. Class diagram of fractal agents

Each fractal agent has been modelled by using two types of UML models, which are a class diagram and an activity diagram. A class diagram is used to describe the types of objects (or classes) that are used within an object-oriented system, and defines the types of relationships between them. An activity diagram is used for defining specific activities and state transitions for corresponding classes. For example, Figure 6 illustrates the activity diagram of GFA (Goal-Formation Agent).

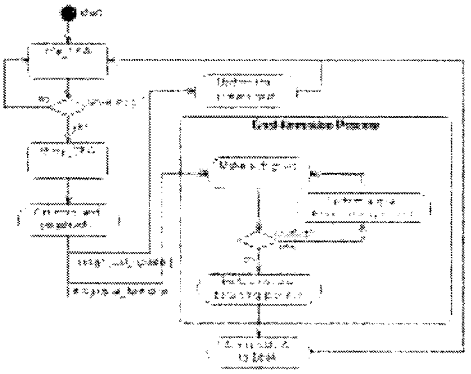

Figure 6. Activity diagram of a GFA

3. Goal-formation in Manufacturing Systems

3.1 Literature Review

In order to deal with multiple objectives or goals in the system, goal programming (GP) models have been used. The GP model is useful for a decision maker (DM) to consider simultaneously several objectives in finding a set of acceptable solutions. Especially, hierarchical optimization has proved to have several significant advantages for the solution of complex optimization problems over conventional methods (Haimes et al. [1990]). However, it is very difficult for a decision maker to find out what attainments are desired for each objective function even though goals and constraints are defined precisely in a standard GP formulation.

To incorporate uncertainty and imprecision into the formulation, several approaches are used to reformulate the GP models such as probability distribution, penalty function, fuzzy numbers and various types of thresholds (Inuiguchi and Kume [1991] and Romero [1984]). Applying fuzzy set theory into GP has the advantage of allowing for the vague aspirations of a DM, which can be qualified by some natural language terms or vague phenomena. For modeling goals with imprecise nature, Martel and Aouni[1998] have reformulated the standard GP model (Charnes and Cooper [1961]) by introducing the satisfaction degrees as a function of the goal deviations in the objective function of the model. Also, Narasimhan[1980] had initially proposed fuzzy goal programming (FGP) by using fuzzy membership functions to specify imprecise aspiration levels of the goals in a fuzzy environment. The FGP formulation has been used in widespread applications of various fields.

The importance and priorities of the goals, which are relatively different from each other, should be considered in the FGP problem, so that more important goal can be expected to be achieved ahead. In the FGP formulation, linguistic variables, such as "very important" and "moderately important", can be used to describe the fuzzy weights of the goals (Narasimhan [1980]). By using the reformulation of FGP, an additive model can be constructed to efficiently resolve the problem with different importance levels and with preemptive priorities.

During propagating goals into several sub-goals, conflicts may arise between sub-goals generated as two or more individuals are involved in solving a particular problem. Adler et al. [1990] have pointed out many tasks of agents regarding this topic in their book, titled "Conflict-resolution strategies for non-hierarchical distributed agents". To maintain an organizational structure of agents, it is necessary to control and predict the communication of the agents in the system. Therefore, an efficient communication scheme is required to resolve the conflicts of goals.

3.2 Goal-Formation Process (GFP) in the FrMS

In order to propagate top-level goal into several sub-goals, the FrMS mainly has three sub-processes including goal-generating process (GGP), goal-harmonizing process (GHP), and goal-balancing process (GBP). After being performed all processes mentioned above, many tasks are generated and conducted by individual agents according to their goals. This is referred to as "goal-achieving process". Finally, the FrMS assesses the system goal (top-level goal) in a bottom-up fashion, i.e., the assessment of the lowest goals is first performed, then higher-level goals are assessed based on the result of sub-goal assessment accordingly. The result of the assessment of individual goals is iteratively affect fractals' current goals while they are achieving goals during the whole lifetime. The GFP is handled mainly by GFA (Goal-Formation Agent) among several fractal agents. Figure 7 briefly illustrates the goal-oriented operations in the FrMS. GFAs in each fractal perform GFP (GGP, GHP, and GBP) and goal assessment.

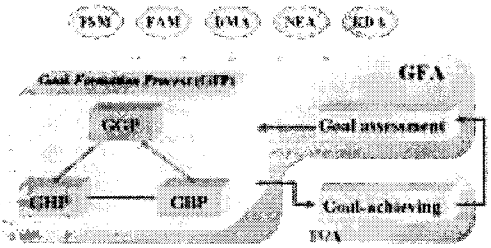

Figure 7. Goal-oriented operations in the FrMS

3.2.1 Goal-Generating Process (GGP)

When a fractal has one or more than one sub-fractals, the goal-generating process (GGP) is required to make sub-goals. This process is initially triggered by the arrival of a system goal (the highest-level goal) from the outside. In order to generate sub-goals, a fractal empowers GFA to perform GGP by cooperating with other agents including FSM (Fractal Status Manager), FAM (Fractal Address Manager), DMA (Decision-Making Agent), and KDA (Knowledge Database Agent). The highest goal of a system can be regarded as integrated form of sub-goals as described in Equation (1).

$$G_0 = g_1 \oplus g_2 \oplus \cdots \oplus g_n \qquad (1)$$

The fractal, located in the top of the fractal architecture (level-0), first subdivides the system goal (G_0) into several fuzzified sub-goals (g_1, g_2, $\cdots$, g_n) based on the functional requirements. The ability and current status of sub-fractals are analyzed before functional decomposition of the goal with the help of FSM and FAM. Then, the highest goal of the system can be regarded as a set of sub-goals, and each sub-goal is iteratively decomposed into sub-goals again as illustrated in Equation (2).

$$g_{j,k}^{i} = \{g_{k,1}^{i+1}, g_{k,2}^{i+1}, \cdots, g_{k,n}^{i+1}\} \qquad (2)$$

where, i - level of the goal
j - goal indicator of the parent goal
k - goal indicator of the current goal
n - total number of sub-goals
(i.e., the number of sub-fractals)

Each goal is fuzzified during the GGP to allow the vague aspirations, and prioritized according to the importance and urgency. The special type of fuzzy values, namely triangular fuzzy numbers (T-numbers) with piece-wise linear membership functions is considered in this paper. Assume that $G - \tilde{T}$, fuzzy goals can be represented as follows;

$$\tilde{T} = \{\tilde{g}^i_{(j,k)} : \tilde{g}^i_{(j,k)} = (g^i_{(j,k)}, \underline{g^i_{(j,k)}}, \overline{g^i_{(j,k)}}),\ \underline{g^i_{(j,k)}} \ge 0,\ \overline{g^i_{(j,k)}} \ge 0\} \quad (3)$$

The membership function $\mu_{\tilde{g}^i_{(j,k)}}$ of $\tilde{g}^i_{(j,k)}$ is defined as follows (see Figure 8);

$$\mu_{\tilde{g}^i_{(j,k)}}(t) = \begin{cases} \max\left(0, 1 - \dfrac{g^i_{(j,k)} - t}{\underline{g^i_{(j,k)}}}\right) & \text{for } t \le g^i_{(j,k)} \\ \max\left(0, 1 - \dfrac{g^i_{(j,k)} - t}{\overline{g^i_{(j,k)}}}\right) & \text{for } t > g^i_{(j,k)} \end{cases} \quad (4)$$

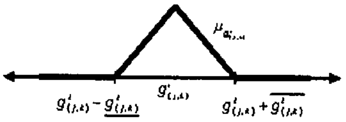

Figure 8. Membership function of a fuzzy goal

In order to develop a FGP formulation, goal functions are required. This paper only concerns linear fuzzy goal functions for enhanced computability of each agent. Detailed descriptions on a FGP formulation are omitted in this paper, because it is not included in our research boundary. After the GGP, each goal is coordinated with each other via GHP and GBP if necessary before being achieved.

3.2.2 Goal-Harmonizing Process (GHP)

The goal-harmonizing process (GHP) is the process to eliminate or reduce conflicts and interferences of goals. An autonomous agent is naturally given control over its own actions with utility functions while pursuing its own interests. This usually occurs conflicts between fractal's individual goals. When a conflict occurs in the system, it can be resolved either through direct negotiation between the conflict participants or through a third party, referred to as the "mediator" (Sycara [1987]). Efficient and effective communication can provide an improved robustness and a balanced goal operation to the system. Therefore, resolution of conflicts is an important procedure to coordinate the systems operated by an intelligent and autonomous agents.

To negotiate among agents, the Contract Net Protocol (CNP) is still widely used, which was proposed by Smith[1980]. However, the CNP is somewhat expensive in terms of network bandwidth. To efficiently distribute the communication load between fractals or agents, a mobile agent-based negotiation protocol (MANPro) was proposed (Shin et al. [2001]). Under the scheme of MANPro, the clone of an agent travels over target agents to negotiate with them if necessary. The negotiation in MANPro is processed following four steps; 1) preparation, 2) cloning, 3) traveling and evaluation, and 4) awarding. Figure 9 illustrates the negotiation process in the FrMS where MANPro is applied.

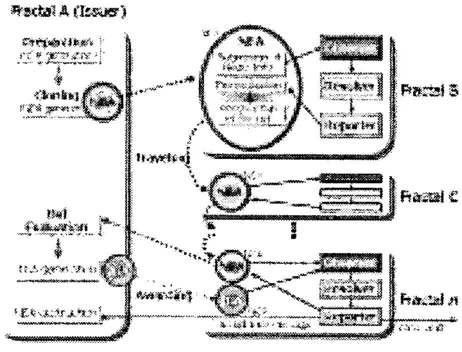

Figure 9. Negotiation Process applying MANPro

The GFA deal with GHP by the help of NEA and DMA. There are three types of resolving cases. The first case is that the conflict is resolved by a single fractal. When the functional areas of each goal, generated by the GGP, are overlapped, i.e., a goal is interfered with others regarding the allowable deviations (or tolerance limit) from aspiration level, the fractal identifies such goals and coordinates them. The second case occurs when a fractal makes several sub-goals and propagates them to sub-fractals without noticing any conflict between sub-goals. However, the conflicts might occur during the goal-achieving status because of the change of environmental situation. In this case, the conflict participants or fractals directly negotiate with each other to resolve the conflicts. Direct negotiation between sub-fractals can be possible when fractals have the same parent. In other words, the level of the goal and the goal indicator of the parent goal (i and j in Equation (2), respectively) should be same to resolve the conflicts of goals. The last case occurs when conflict goals spread over different goal levels, or conflict goals have a different parent goal even though they are in the same goal level. In this case, information of the conflict is encrypted into a NEA and delivered to the fractal with conflict goals with the help of FSM and FAM in several interim fractals. After the negotiation between fractals, the GFAs of conflict fractals coordinate their goals in accordance with the DMAs' decision.

3.2.3 Goal-Balancing Process (GBP)

Balancing of goals in manufacturing systems is very difficult to consider because of goals' variety and specialized performance measures etc. As the goal is iteratively decomposed based on functional requirements, the goals in the lowest level becomes similar to an operational objective or a short time schedule. In this regard, many researchers have endeavored to maximize objective satisfaction by developing various performance indicators. The main problem in manufacturing system, however, is not thought to satisfy independent objectives at the same time, but to define a satisfactory compromise between partially conflicting objectives (Smith [1992] and van der Pluym [1990]) as has often been underlined.

In any case, parameterable or quantifiable indicators representing the goals and a goal system is to be developed for balancing and assessing them explicitly (the development of such indicators has not been completed yet). As an example of such indicators, Grabot[1998] proposed a general structure of the short-term manufacturing objectives by using a hierarchical structure as illustrated in Figure 10. Each objective has several indicators but omitted in the figure because of a graphic simplicity.

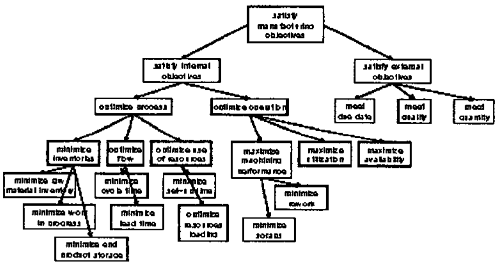

Figure 10. General structure of operational objectives (from Grabot[1998])

This paper adopts a neural network (NN) approach to balance goals. A NN may provide a way to imitate a complex transfer function that cannot be analytically identified. Since it is difficult and ambiguous to analytically describe the relations between objectives (or goals) and quantifiable indicators, the use of NN for GBP in the FrMS may have following benefits;

- A priori analytical function linking inputs to outputs is not necessary to be chosen.
- The parameterization of the NN may be performed without any knowledge of the NN approach, i.e., the technique allowing us to imitate the expert aggregation is transparent for the user.
- The method is iterative so that new examples can replace or complete the first ones in order to increase the adequacy of the system to the customer's requirements.

For the bottom level fractals, i.e., the fractals directly controlling manufacturing resources (e.g., CNC machine, turning machine, robots, material handlers, etc.), their individual goal is much similar to the job schedule. Since the goals of the fractal are delivered from the upper-level fractal in a fuzzified manner, the GFA in the bottom level should defuzzify the individual goal into explicit forms, i.e., executable goals, through the GBP. In order to defuzzify the fuzzified goals, the GFA generates several alternatives within the range of the tolerance limits. Then the DMA chooses the best one, referring to the situation of the fractal and higher goals.

To keep the balance of the goals of the bottom level fractals, the workload of each fractal must be calculated first, which is mainly burdened by 1) the communication load occurred when the fractal negotiates with other fractals, and 2) serial communication load occurred when the fractal controls equipment. The communication load burdened between fractals is very difficult to expect because the system changes dynamically. On the other hand, the expectation of the serial communication load is relatively easy if the control messages are completely specified. To do so, equipment has been classified into four major types including material processor (*MP*), material handler (*MH*), material transporter (*MT*), and buffer storage (*BS*). Each type is refined into several sub-types again. After that, the information and messages for controlling equipment have been defined and classified into two major categories based on the equipment types; 1) durable information and messages, and 2) instantaneous information and messages. Quntification of defined information and messages, the workload of the bottom level fractal can be analytically calculated. As a consequence, the GBP can be facilitated by using such informations.

4. Concluding Remarks

The FrMS has fractal-specific characteristics which can be effectively used in a dynamic and distributed system. Among several characteristics, this paper focused on the goal-oriented nature. The fractals in the FrMS autonomously evolve their goals by using goal-formation process (GFP) proposed in this paper. The GFP is iteratively performed by three sub-processes, which are goal-generating process (GGP), goal-harmonizing process (GHP), and goal-balancing process (GBP). By using the GFP framework proposed in this paper, the FrMS can get enhanced autonomy while handling multiple and complicated goals.

However, a goal-achieving mechanism and goal-assessment scheme should be additively integrated into the dynamic model of GFA in order to develop concrete GFP framework.

Acknowledgements

This research was supported in part by the grant No. 2001-1-31500-005-1 from the Basic Research Program of the Korea Science & Engineering Foundation and the BK 21 Project in 2003. The authors would like to express their gratitude for the support.

References

[1] Adler, M. R., Davis, A. B., Weihmayer, R., and Worrest, R. W., Conflict-resolution strategies for nonhierarchical distributed agents, 1990.
[2] Brussel, H. V., Wyns, J., Valckenaers, P., Bongaerts, L., and Peeters, P., Reference architecture for holonic manufacturing systems: PROSA, Computers in Industry 37 (3), 1998, pp. 255-274.
[3] Chang, T. C., Wysk, R. A., and Wang, H. P., Computer Integrated Manufacturing, Prentice-Hall, 1998.
[4] Charnes, A. and Cooper, W. W., Management Models and Industrial Applications of Linear Programming, Wiley, New York, 1961.
[5] Cho, H., An Intelligent Workstation Controller for Computer Integrated Manufacturing, Ph.D. dissertation, Texas A&M University, USA, 1993.
[6] Grabot, B., Objective satisfaction assessment using neural nets for balancing multiple objectives, International Journal of Production Research, 36 (9), 1998, pp. 2377-2395.
[7] Haimes, Y. Y., Tarvainen, K., Shima, T., and Thadanthil, J., Hierarchical-multiobjective analysis of large-scale systems, Hemisphere, New York, 1990.
[8] Inuiguchi, M. and Kume, Y., Goal programming problems with interval coefficients and target intervals, European Journal of Operational Research, 52 (3), 1991, pp. 345-360.
[9] Martel, J. M. and Aouni, B., Diverse formulations of the impresice goal programming model, Journal of Global Optimization, 12, 1998, pp. 127-138.
[10] Mettala, E. G., Automatic Generation of Control Software in Computer Integrated Manufacturing, Ph.D. dissertation, The Penn State University, USA, 1989.
[11] Narasimhan, R., Goal programming in a fuzzy environment, Decision Sciences, 11, 1980, pp. 325-336.
[12] Okino, N., Bionic manufacturing systems, Proceedings of the CIRP Seminar on Flexible Manufacturing Systems Past-Present-Future, Peklenik, J. (ed.), Bled, Slovenia, 1993, pp. 73-95.
[13] Romero, C., A note: Effects of five-side penalty functions in goal programming, Omega, 12, 1984, 333.
[14] Ryu, K. and Jung, M., Dynamic Modeling of Fractal-specific Characteristics in Fractal Manufacturing Systems, Proceedings of the 35th CIRP-International Seminar on Manufacturing Systems, Seoul, Korea, 2002, pp. 444-450.
[15] Ryu, K., Shin, M., and Jung, M., A Methodology for Implementing Agent-based Controllers in the Fractal Manufacturing System, Proceedings of 5th Conference on Engineering Design & Automation, Las Vegas, NV, 2001, pp. 91-96.
[16] Ryu, K., Shin, M., Kim, K., and Jung, M., Intelligent Control Architecture for Fractal Manufacturing System, Proceedings of 3rd Asia-Pacific Conference on Industrial Engineering and Management Systems, Hong Kong, 2000, pp. 594-598
[17] Seidel, D., Hopf, M., Prado, J. M., Garcia-Herreros, E., Strasser, T. D., Christensen, J. H., and Oblak, J. M., HMS-Strategies, The Report of HMS Consortium, 1994.
[18] Shin, M., Ryu, K., and Jung, M., A novel negotiation protocol for agent-based control architecture, Proceedings of 5th conference on Engineering Design & Automation, Las Vegas, NV, 2001, pp. 700-705.
[19] Smith, J., A Formal Design and Development Methodology for Shop Floor Control in Computer Integrated Manufacturing, Ph.D. dissertation, The Penn State University, USA, 1992.
[20] Smith, R. G., The contract net protocol: high-level communication and control in a distributed problem solver, IEEE Trans. on Computers, 29 (12), 1980, pp. 1104-1113.
[21] Smith, S. F, Knowledge-base production management: approaches, results and prospectives, Production Planning and Control, 3, 350-380.
[22] Son, Y., Simulation Based Shop Floor Control: Automatic Model Generation and Control Interface, Ph.D. dissertation, The Penn State University, USA, 2000.
[23] Sycara, K. P., Multiagent compromise via negotiation, in: Distributed artificial intelligence, II, Pitman, London, 1987, pp. 119-137.
[24] Tharumarajah, A., Wells, A. J., and Nemes, L., Comparison of the bionic, fractal and holonic manufacturing system concepts. International Journal of Computer Integrated Manufacturing, 9, 1996, pp. 217-22.
[25] Tirpak, T. M., Daniel, S. M., LaLonde, J. D., and Davis, W. J., A note on a fractal architecture for modeling and controlling flexible manufacturing systems, IEEE Transactions on Systems, Man, and Cybernetics, 22, 1992, pp. 564-567.
[26] Ueda, N., A concept for bionic manufacturing systems based on DNA-type information, Human Aspects in Computer Integrated Manufacturing, 1992, pp. 853-863.
[27] Van der Pluym, B., Knowledge-based decision making for job-shop scheduling, International Jurnal of Integrated Manufacturing, 3, 1990, pp. 354-363.
[28] Warnecke, H. J., The Fractal Company: a Revolution in Corporate Culture, Springer-Verlag, Berlin, 1993.
[29] Wong, J., Helmer, G., Naganathan, V., Polavarapu, S., Honavar, V., and Miller, L., SMART mobile agent facility. Journal of Systems and Software, 56, 2001, pp. 9-2.