DESCRIPTION

Title of Invention:

CONTROL APPARATUS, CONTROL METHOD, AND CONTROL PROGRAM

Technical Field

[0001] The present invention relates to a technique that arbitrates, for example, contending requests outputted from a plurality of software to a device.

Background Art

[0002] As the function and performance of consumer electronics such as refrigerators, air-conditioners, cooking heaters, and dryers improve due to the technological advancement, the volume of development of embedded software increases.

In the older days when the consumer electronics did not carry microcomputers, device control was achieved by a hardware circuit. With the advent of the microcomputers, however, device control by embedded software has become the mainstream. In the transition of the control from the hardware circuit to the embedded software, the hard wired logic of the hardware circuit was replaced by a device-controlling program.

[0003] When the program is to replace the hard wired logic, numerous instructions are generated according to the procedure of the machine language in order to convert the functions realized by the hard wired logic into the machine language directly. Embedded software thus created is less structure-designed than the embedded structure created based on a logical program design method from the beginning. The source code of such embedded software tends to be a very complicated one called spaghetti code. It is thus difficult to conduct maintenance such as identifying the portion where a problem occurs and to expand the function for the purpose of improvement.

[0004] At present, the embedded software is changing from the above-described program written in the machine language to a program written in a general computer-oriented language. When creating the embedded software, a program portion called a device driver, which transmits a simple operation instruction to the device directly is created independently of other program portions and is used for several devices. This improves the development efficiency than in the development of the older embedded software using the machine language.

[0005] Numerous functions are required of the product and their number increases more and more as the product function improves. Diverse requests outputted from the numerous functions may sometimes contend with each other when actuating the device finally. It is required of the embedded software to arbitrate the contending requests appropriately, but it is difficult to create the arbitration function, which is a problem.

[0006] In general, a developer develops embedded software by creating programs for the respective functions based on a product requirement specification. The developer creates a program that realizes the required functions, in such a manner that the program also includes a program describing a condition with which the requests for the device are arbitrated.

If the product has only a small number of functions, the program may be created comparatively easily this way. However, for a product in which numerous functions have diverse requests for the device, if the requests contend, the condition on the determination for validating and invalidating the requests of the functions becomes very complicated.

In that case, only the developer who initially developed the embedded software can understand the structure of the program that realizes the complicated condition, and it is very difficult for other developers to revise the program. If other developers

revise the program without a sufficient understanding of the program that realizes the condition, bugs may sometimes increase due to the following reason. Since the numerous functions are closely associated to one device, if the program is revised concerning only one function, the revision may influence even a function that seems irrelevant, without the developer knowing it.

[0007] As a method of developing embedded software, efficient program design methods such as object-oriented design and structured design have been contrived. Based on these design methods, a software platform for development, a middleware/software automatic creation tool, and the like have been developed.

[0008] As an example, Patent Literature 1 describes design modeling in which the function of embedded software is divided as control filter components, and the function-divided control filter components are arranged with their designated priorities.

[0009] As another example, Non-Patent Literature 1 describes design modeling in which management is divided into three major roles: management of the device status transition; management of contention solution in using the device; and management of the time constraints imposed on the respective processes.

Citation List

Patent Literature

[0010] Patent Literature 1: JP 10-78809

Non-Patent Literature

[0011] Non-Patent Literature 1: Masami IWAHASHI, Naruki MITSUDA, Tsuneo AJISAKA, and Tsuyoshi NAKAJIMA: "Proposal on an Object-oriented Development Technique for Embedded Systems using Autonomic Objects and Contention Resolution Methods", IPSJ (Information Processing Society of Japan) SIG Technical Report, Vol. 2009-SE-164 No. 11.

Summary of Invention

Technical Problem

[0012] Even if the design modeling described in Patent Literature 1 or Non-Patent Literature 1 is employed, a high-order program portion which realizes the function and a low-order program portion which controls the device are closely connected to each other. Consequently, structured program design becomes difficult to perform, the software becomes complicated, and the maintainability, reusability, and expandability suffer.

It is an object of the present invention to simplify the processing structure that arbitrates contending requests.

Solution to Problem

[0013] A control apparatus according to the present invention which arbitrates contending operation requests outputted from a plurality of APPs, APP standing for application program, and controls a device, includes:

a request input part which supplies, as input, the operation requests outputted from the APPs, each to a phase predetermined for a corresponding APP among a plurality of phases arranged in a fixed order;

an arbitration information storage part which stores, for each phase, arbitration information;

a request generating part which generates an operation request, being a result of arbitrating a contention of the operation requests inputted, in the phases sequentially starting with the first phase and in accordance with the arbitration information stored in the arbitration information storage part, and supplies, as input, the operation request generated, to a next phase; and

a device control part which controls an operation of the device in accordance

with the operation request generated by the request generating part in the last phase.

Advantageous Effects of Invention

[0014] The control apparatus according to the present invention need not include a function of arbitrating contending requests in the application programs, and can separate the application programs from the function of arbitrating the contending requests. In particular, the control apparatus according to the present invention arbitrates the requests sequentially through a plurality of phase. Thus, the processing structure that arbitrates the contending requests can be simplified.

Brief Description of Drawings

[0015] Fig. 1 is a configuration diagram of a control apparatus 100 according to Embodiment 1.

Fig. 2 is a configuration diagram of a middleware part 20 according to Embodiment 1.

Fig. 3 is a flowchart showing the operation of the middleware part 20 according to Embodiment 1.

Fig. 4 shows a case in which the control apparatus 100 is built in a refrigerator.

Fig. 5 is a table showing arbitration information concerning phase 1 according to Embodiment 1.

Fig. 6 is a table showing arbitration information concerning phase 2 according to Embodiment 1.

Fig. 7 is a configuration diagram according to Embodiment 2.

Fig. 8 is a flowchart showing the operation of a middleware part 20 according to Embodiment 2.

Fig. 9 is a table showing arbitration information concerning phase according to Embodiment 2.

Fig. 10 is a configuration diagram of a control apparatus 100 according to Embodiment 3.

Fig. 11 is a flowchart showing the operation of a middleware part 20 according to Embodiment 3.

Fig. 12 is a table showing arbitration information concerning phase 1 according to Embodiment 3.

Fig. 13 is a configuration diagram of a control apparatus 100 according to Embodiment 4.

Fig. 14 is a configuration diagram of a middleware part 20 according to Embodiment 4.

Fig. 15 is a flowchart showing the operation of the middleware part 20 according to Embodiment 4.

Fig. 16 is a table showing arbitration information concerning phase 1 according to Embodiment 4.

Fig. 17 shows a case in which the control apparatus 100 is built in an air-conditioner.

Fig. 18 is a chart showing secondary requests outputted in the case shown in Fig. 17.

Fig. 19 shows a case in which a general control apparatus is built in an air-conditioner.

Fig. 20 is an explanatory diagram of a DC (direct current) motor-driven automated cart.

Fig. 21 shows a case in which the control apparatus 100 is built in the automated cart shown in Fig. 18.

Fig. 22 shows an example of the hardware configuration of a control apparatus

100 according to the above embodiments.

Description of Embodiments

[0016] Embodiment 1.

Fig. 1 is a configuration diagram of a control apparatus 100 according to Embodiment 1.

The control apparatus 100 controls a consumer electronics product. The control apparatus 100 is provided with an application part 10, a middleware part 20, a device control part 30, and a device driver 40.

[0017] The application part 10 is an application program (to be abbreviated as APP hereinafter) that realizes some function and is provided with a basic operation execution part 11 and a plurality of function execution parts 12 (function execution parts 12A to 12E in this case).

The basic operation execution part 11 is an APP that realizes the basic operation of the consumer electronics product. When the consumer electronics product is ON, the basic operation execution part 11 operates regardless of the operation by the user, the external atmosphere, and the like.

The function execution part 12 is an APP that realizes an additional function of the consumer electronics product. The function execution part 12 is provided for each additional function, and operates in accordance with the operation by the user, the external environment, and the like.

[0018] The middleware part 20 acquires device operation requests (to be referred to as primary requests hereinafter) from the basic operation execution part 11 and the respective function execution parts12, and generates and outputs a new operation request (to be referred to as a secondary request hereinafter) obtained by arbitrating the contending primary requests.

The middleware part 20 has a plurality of phases (phases 1 to 4 in this example) arranged in a fixed order. The middleware part 20 supplies, as input, the primary requests outputted from the basic operation execution part 11 and the function execution parts 12, each to a phase. The middleware part 20 repeats arbitrating the inputted primary requests, in the phases sequentially starting with an upper phase and supplying, as input, the arbitrated primary request to the next phase, to finally generate a secondary request.

[0019] In accordance with the secondary request generated by the middleware part 20, the device control part 30 generates a control parameter to be supplied to the device driver 40. The device control part 30 supplies the generated control parameter to the device driver 40, thereby controlling the device.

[0020] In accordance with the control parameter generated by the device control part 30, the device driver 40 actuates the device (hardware).

[0021] Fig. 2 is a configuration diagram of the middleware part 20 according to Embodiment 1.

The middleware part 20 is provided with a request input part 21, an arbitration information storage part 22, and a request generating part 23.

The request input part 21 supplies, as input, the primary request outputted from the basic operation execution part 11, to the first phase 1. Also, the request input part 21 supplies, as input, the primary request outputted from each function execution part 12, to a phase predetermined for a corresponding function execution part 12, among the plurality of phases.

The arbitration information storage part 22 stores, for each phase, arbitration information used for arbitrating the contention of the primary requests, in a storage device. The arbitration information indicates the priorities of the respective primary

requests.

The request generating part 23 selects a primary request for which the arbitration information indicates the highest priority, from among the inputted primary requests, in the phases sequentially starting with the first phase. In a phase other than the last phase, the request generating part 23 supplies, as input, the selected primary request to the next phase. In the last phase, the request generating part 23 outputs the selected primary request to the device control part 30 as the secondary request.

[0022] Fig. 3 is a flowchart showing the operation of the middleware part 20 according to Embodiment 1.

When the consumer electronics product is turned on, the control apparatus 100 is actuated, and the process is started.

The request input part 21 then supplies, as input, the primary request outputted from the basic operation execution part 11, to phase 1, and supplies, as input, the primary requests outputted from each function execution part 12, to predetermined phases (S11).

Subsequently, the request generating part 23 initializes a variable i representing a phase, to 1 (S12). The request generating part 23 then, for phase i, selects the highest-priority primary request from among the inputted primary requests (S13). The request generating part 23 checks whether or not phase i is the last phase (S14). If phase i is not the last phase, the request generating part 23 supplies, as input, the selected primary request to the next phase i+1 (S15), adds 1 to the variable i (S16), and puts the process back to S13. If phase i is the last phase, the request generating part 23 outputs the selected primary request to the device control part 30 as the secondary request (S17).

When a predetermined period of time elapses after outputting the secondary

request, the request generating part 23 executes the process again starting with S11.

[0023] Fig. 4 shows a case in which the control apparatus 100 is built in a refrigerator.

In general, the refrigerator is provided with a compressor (an example of the device). By changing the operation speed (rotation speed rpm or rotation frequency Hz) of the compressor, the temperature in the refrigerator can be controlled. A case will now be described where the control apparatus 100 controls the operation speed of the compressor of the refrigerator.

Assume that the operation speed of the compressor speed ranges from 0th speed of a stop state to the highest 10th speed.

[0024] When the refrigerator is turned on, it carries out the operation of cooling the inside of the refrigerator continuously. This operation is executed by the basic operation execution part 11.

The refrigerator has a rapid-cooling function of cooling the inside of the refrigerator rapidly and an ice-making function of making ice quickly. The rapid-cooling function and the ice-making function are executed when a rapid-cooling button and an ice-making button are depressed, and are ended when predetermined conditions are satisfied. The refrigerator also has weak- and strong-protection functions of suppressing the operation of the compressor in order to prevent a trouble of the compressor and the like. The weak- and strong-protection functions are executed when predetermined respective protecting conditions are satisfied, and are ended when the protecting conditions cease to be satisfied. The rapid-cooling function, the ice-making function, and the respective protecting functions are executed by the corresponding function execution parts 12. Note that a function execution part 12A executes the rapid-cooling function, a function execution part 12B executes the ice-making function, a function execution part 12C executes the weak-protection

function, and a function execution part 12D executes the strong-protection function.

A rapid-cooling request as a primary request outputted from the function execution part 12A and an ice-making request as a primary request outputted from the function execution part 12B are inputted to phase 1. A weak-protection request as a primary request outputted from the function execution part 12C and a strong-protection request as a primary request outputted from the function execution part 12D are inputted to phase 2.

A basic request as a primary request outputted from the basic operation execution part 11 requests 4th-speed operation. The rapid-cooling request, the ice-making request, the weak-protection request, and the strong-protection request respectively request 7th-speed operation, 5th-speed operation, 2nd-speed operation, and 0th-speed operation.

[0025] Fig. 5 is a table showing the arbitration information concerning phase 1 according to Embodiment 1.

The arbitration information shown in Fig. 5 represents the priorities of the basic request, rapid-cooling request, and ice-making request. In Fig. 5, the basic request has the lowest priority, the rapid-cooling request has the second-lowest priority, and the ice-making request has the highest priority.

[0026] Fig. 6 is a table showing the arbitration information concerning phase 2 according to Embodiment 1.

The arbitration information shown in Fig. 6 represents the priorities of the primary request, weak-protection request, and strong-protection request generated in phase 1. In Fig. 6, the primary request generated in phase 1 has the lowest priority, the weak-protection request has the second-lowest priority, and the strong-protection request has the highest priority.

[0027] An example of the process will be described with reference to the flowchart of Fig. 3.

A case will be described where the operation of the refrigerator is started (stage 1), the rapid-cooling button is depressed some time after that (stage 2), the ice-making button is depressed during the rapid-cooling operation (stage 3), and the weak-protection function is actuated during the rapid-cooling and ice-making operations (stage 4).

[0028] <Stage 1>

The refrigerator is turned on, and the control apparatus 100 is actuated. Then, the request input part 21 supplies, as input, the basic request as the primary request outputted from the basic operation execution part 11, to phase 1 (S11). At this time point, no additional function is active, and no other primary request is inputted.

The request generating part 23 initializes the variable i to 1 (S12) and arbitrates the contention concerning phase 1 (S13). Since there is no other primary request than the basic request outputted from the basic operation execution part 11, a contention of the primary requests does not occur. Hence, the request generating part 23 selects the basic request outputted from the basic operation execution part 11. Since phase 1 is not the last phase (NO in S14), the request generating part 23 supplies, as input, the basic request selected in S13 to phase 2 (S15). Then, the request generating part 23 adds 1 to the variable i to obtain 2, and puts the process back to S13 (S16).

The request generating part 23 arbitrates a contention concerning phase 2 (S13). Since there is no other primary request than the basic request inputted from phase 1, a contention of the primary requests does not occur. Hence, the request generating part 23 selects the basic request inputted from phase 1. Since phase 2 is the last phase (YES in S14), the request generating part 23 outputs the basic request

selected in S13 to the device control part 30 as the secondary request (S17).

In brief, at this point, the basic request is directly outputted to the device control part 30, and the compressor operates at the 4th speed.

[0029] <Stage 2>

After that the rapid-cooling button is depressed. Then, the request input part 21 supplies, as input, the basic request outputted from the basic operation execution part 11 and the rapid-cooling request outputted from the function execution part 12A, to phase 1 (S11).

The request generating part 23 initializes the variable i to 1 (S12), and arbitrates the contention concerning phase 1 (S13). Since the basic request and the rapid-cooling request are inputted, the request generating part 23 arbitrates the contention in accordance with the arbitration information indicated in Fig. 5. As a result, a rapid-cooling request having a higher priority is selected. Since phase 1 is not the last phase (NO in S14), the request generating part 23 supplies, as input, the rapid-cooling request selected in S13 to phase 2 (S15). Then, the request generating part 23 adds 1 to the variable i to obtain 2, and puts the process back to S13 (S16).

The request generating part 23 arbitrates the contention concerning phase 2 (S13). Since there is no other primary request than the rapid-cooling request inputted from phase 1, a contention of the primary requests does not occur. Hence, the request generating part 23 selects the rapid-cooling request inputted from phase 1. Since phase 2 is the last phase (YES in S14), the request generating part 23 outputs the rapid-cooling request selected in S13 to the device control part 30 as the secondary request (S17).

In brief, at this point, the rapid-cooling request is outputted to the device control part 30, and the compressor operates at the 7th speed.

[0030] <Stage 3>

After that the ice-making button is depressed during the rapid-cooling operation. Then, the request input part 21 supplies, as input, the basic request outputted from the basic operation execution part 11, the rapid-cooling request outputted from the function execution part 12A, and the ice-making request outputted from the function execution part 12B, to phase 1 (S11).

The request generating part 23 initializes the variable i to 1 (S12), and arbitrates the contention concerning phase 1 (S13). Since the basic request, the rapid-cooling request, and the ice-making request are inputted, the request generating part 23 arbitrates the contention in accordance with the arbitration information indicated in Fig. 5. As a result, the ice-making request having a higher priority is selected. Since phase 1 is not the last phase (NO in S14), the request generating part 23 supplies, as input, the ice-making request selected in S13 to phase 2 (S15). Then, the request generating part 23 adds 1 to the variable i to obtain 2, and puts the process back to S13 (S16).

The request generating part 23 arbitrates the contention concerning phase 2 (S13). Since there is no other primary request than the ice-making request inputted from phase 1, a contention of the primary requests does not occur. Hence, the request generating part 23 selects the ice-making request inputted from phase 1. Since phase 2 is the last phase (YES in S14), the request generating part 23 outputs the ice-making request selected in S13 to the device control part 30 as the secondary request (S17).

In brief, at this point, the ice-making request is outputted to the device control part 30, and the compressor operates at the 5th speed.

[0031] <Stage 4>

After that the weak-protection function is actuated during the ice-making

operation. Then, the request input part 21 supplies, as input, the basic request outputted from the basic operation execution part 11, the rapid-cooling request outputted from the function execution part 12A, and the ice-making request outputted from the function execution part 12B, to phase 1 (S11), and supplies, as input, the weak-protection function outputted from the function execution part 12C, to phase 2 (S11).

The request generating part 23 initializes the variable i to 1 (S12), arbitrates the contention concerning phase 1 in the same manner as in stage 3, and supplies, as input, the rapid-cooling request to phase 2 (S13-S15). Then, the request generating part 23 adds 1 to the variable i to obtain 2, and puts the process back to S13 (S16).

The request generating part 23 arbitrates the contention concerning phase 2 (S13). Since the ice-making request inputted from phase 1 and the weak-protection request outputted from the function execution part 12C are inputted, the request generating part 23 arbitrates the contention in accordance with the arbitration information indicated in Fig. 6. As a result, the weak-protection request having a higher priority is selected. Since phase 2 is the last phase (YES in S14), the request generating part 23 outputs the weak-protection request selected in S13 to the device control part 30 as the secondary request (S17).

In brief, at this point, the weak-protection request is outputted to the device control part 30, and the compressor operates at the 2nd speed.

[0032] As described above, the control apparatus 100 according to Embodiment 1 has the middleware part 20 between the application part 10 and the device control part 30. The middleware part 20 arbitrates the contending operation requests outputted from the application part 10 to the device.

Hence, in the application part 10, the basic operation execution part 11 and the

function execution parts 12 which realize the respective functions need to be implemented only with the functions that are to be realized, and concerning device control, need to output only an operation request. The function of arbitrating the contending operation requests may be implemented in the middleware part 20 independently of the respective functions.

In particular, regarding the function of arbitrating the contending operation requests, the operation requests outputted from the respective functions are processed stepwise through a plurality of phases. Hence, in each phase, the function of arbitrating the contending operation requests can have a simple configuration. More specifically, in each phase, the function of arbitrating the contending operation requests can be expressed using the priorities shown in Figs. 5 and 6.

As a result, unlike with conventional embedded software, the function parts that realize the respective functions do not include a function of arbitrating the contending operation requests, so that the respective function parts no longer closely relate to each other.

[0033] Embodiment 2.

Embodiment 1 described how in each phase the primary request having the highest priority is selected from among the inputted primary requests. There may be, however, a case where two or more primary requests can be satisfied simultaneously. In the case of the refrigerator described above, the rapid-cooling request and the ice-making request can be realized simultaneously if the speed of the compressor is increased.

Embodiment 2 describes how, in a phase where two or more primary requests can be satisfied simultaneously, a new primary request that realizes two or more primary requests simultaneously is generated instead of selecting the primary request

having the highest priority from among the inputted primary requests.

[0034] Fig. 7 is a configuration diagram of a control apparatus 100 according to Embodiment 2.

The control apparatus 100 according to Embodiment 2 is different from the control apparatus 100 according to Embodiment 1 in that each phase is determined to be either a selection phase or a combination phase.

The selection phase is a phase described in Embodiment 1. That is, in the selection phase, one primary request is selected from among a plurality of inputted primary requests.

The combination phase is a phase of generating a new primary request that realizes two or more primary requests simultaneously. In the combination phase, a new primary request is generated to correspond to the combination of inputted operation requests.

[0035] The configuration of a middleware part 20 according to Embodiment 2 is the same as the configuration of the middleware part 20 according to Embodiment 1 shown in Fig. 2. The middleware part 20 according to Embodiment 2 is different from the middleware part 20 according to Embodiment 2 in arbitration information stored by an arbitration information storage part 22.

Concerning the selection phase, the arbitration information storage part 22 stores the priorities as the arbitration information, in the same manner as in Embodiment 1. Concerning the combination phase, however, the arbitration information storage part 22 stores, as the arbitration information, combination information that determines a new primary request to be generated to correspond to the combination of the inputted primary requests.

If the arbitration information is combination information, a request generating

part 23 generates a new primary request to correspond to the combination of the inputted primary requests.

[0036] Fig. 8 is a flowchart showing the operation of the middleware part 20 according to Embodiment 2.

S21 to S22 are the same as S11 to S12 shown in Fig. 3, and S24 to S27 are the same as S14 to S17 shown in Fig. 3.

In S23, if phase i is a selection phase, the request generating part 23 selects the primary request having the highest priority from among the inputted primary requests. If phase i is a combination phase, the request generating part 23 generates the new primary request to correspond to the combination of the inputted primary requests, in accordance with the combination information.

[0037] A case (a case in which the control apparatus 100 is built in a refrigerator) shown in Fig. 4 will be described.

Note that phase 1 is a combination phase and phase 2 is a selection phase. Arbitration information on phase 1 is arbitration information shown in Fig. 9. The arbitration information shown in Fig. 9 is combination information indicating a new primary request to be generated to correspond to the combination of primary requests outputted from a basic operation execution part 11, a rapid-cooling function, and an ice-making function. Note that a basic request is outputted constantly. Hence, a case where a basic request is not outputted is omitted.

[0038] In stage 1, only the basic request is inputted. Thus, the refrigerator operates at the 4th speed according to the inputted basic request, in the same manner as in Embodiment 1.

[0039] <Stage 2>

After stage 1, the rapid-cooling button is depressed. Then, a request input

part 21 supplies, as input, the basic request outputted from the basic operation execution part 11 and the rapid-cooling request outputted from a function execution part 12A, to phase 1 (S21).

The request generating part 23 initializes the variable i to 1 (S22), and arbitrates the contention concerning phase 1 (S23). Since the basic request and the rapid-cooling request are inputted, the request generating part 23 arbitrates the contention in accordance with the arbitration information indicated in Fig. 9. As a result, a rapid-cooling request corresponding to the basic & rapid-cooling requests is generated. Since phase 1 is not the last phase (NO in S24), the request generating part 23 supplies, as input, the rapid-cooling request generated in S23 to phase 2 (S25). Then, the request generating part 23 adds 1 to the variable i to obtain 2, and puts the process back to S23 (S26).

The request generating part 23 arbitrates the contention concerning phase 2 (S23). Since there is no other primary request than the rapid-cooling request inputted from phase 1, a contention of the primary requests does not occur. Hence, the request generating part 23 selects the rapid-cooling request inputted from phase 1. Since phase 2 is the last phase (YES in S24), the request generating part 23 outputs the rapid-cooling request selected in S23 to a device control part 30 as the secondary request (S27).

In brief, at this point, the rapid-cooling request is outputted to the device control part 30, and the compressor operates at the 7th speed.

[0040] <Stage 3>

After that the ice-making button is depressed during the rapid-cooling operation. Then, the request input part 21 supplies, as input, the basic request outputted from the basic operation execution part 11, the rapid-cooling request

outputted from the function execution part 12A, and the ice-making request outputted from a function execution part 12B, to phase 1 (S21).

The request generating part 23 initializes the variable i to 1 (S22), and arbitrates the contention concerning phase 1 (S23). Since the basic request, the rapid-cooling request, and the ice-making request are inputted, the request generating part 23 arbitrates the contention in accordance with the arbitration information indicated in Fig. 9. As a result, a rapid-cooling & ice-making request corresponding to the basic & rapid-cooling & ice-making requests is generated. Since phase 1 is not the last phase (NO in S24), the request generating part 23 supplies, as input, the ice-making request generated in S23, to phase 2 (S25). Then, the request generating part 23 adds 1 to the variable i to obtain 2, and puts the process back to S23 (S26).

The request generating part 23 arbitrates the contention concerning phase 2 (S13). Since there is no other primary request than the rapid-cooling & ice-making request inputted from phase 1, a contention of the primary requests does not occur. Hence, the request generating part 23 selects the rapid-cooling & ice-making request inputted from phase 1. Since phase 2 is the last phase (YES in S24), the request generating part 23 outputs the rapid-cooling & ice-making request selected in S23 to the device control part 30 as the secondary request (S27).

In brief, at this point, the rapid-cooling & ice-making request is outputted to the device control part 30, and the compressor operates at the 10th speed.

[0041] <Stage 4>

After that the weak-protection function is actuated during the rapid-cooling & ice-making operation. Then, the request input part 21 supplies, as input, the basic request outputted from the basic operation execution part 11, the rapid-cooling request outputted from the function execution part 12A, and the ice-making request outputted

from the function execution part 12B, to phase 1, and supplies, as input, the weak-protection function outputted from a function execution part 12C, to phase 2 (S21).

The request generating part 23 initializes the variable i to 1 (S22), arbitrates the contention concerning phase 1, in the same manner as in stage 3, and supplies, as input, the rapid-cooling & ice-making request to phase 2 (S23-S25). Then, the request generating part 23 adds 1 to the variable i to obtain 2, and puts the process back to S3 (S26).

The request generating part 23 arbitrates the contention concerning phase 2 (S23). Since the rapid-cooling & ice-making request inputted from phase 1 and the weak-protection request outputted from the function execution part 12C are inputted, the request generating part 23 arbitrates the contention in accordance with the arbitration information shown in Fig. 6. As a result, the weak-protection request having a higher priority is selected. Since phase 2 is the last phase (YES in S24), the request generating part 23 outputs the weak-protection request selected in S23 to the device control part 30 as the secondary request (S27).

In brief, at this point, the weak-protection request is outputted to the device control part 30, and the compressor operates at the 2nd speed.

[0042] As described above, the control apparatus 100 according to Embodiment 2 stores the combination information as the arbitration information, so that a new primary request that realizes two or more primary requests simultaneously can be generated.

Hence, while simplifying the function of arbitrating the contending operation requests in the same manner as in Embodiment 1, control can be carried out more flexibly than in Embodiment 1.

[0043] Embodiment 3.

Embodiments 1 and 2 described how the contention in each phase is arbitrated based only on the inputted primary requests. Embodiment 3 describes how a contention is arbitrated based not only on the inputted primary requests but also on the operation status of the device.

[0044] Fig. 10 is a configuration diagram of a control apparatus 100 according to Embodiment 3.

The control apparatus 100 according to Embodiment 3 is different from the control apparatus 100 according to Embodiment 1 or 2 in that each phase is determined to be a selection phase, a combination phase, or a transition phase.

The transition phase is a phase of generating a new primary request to correspond to the operation status of the device and inputted operation requests.

[0045] The configuration of a middleware part 20 according to Embodiment 3 is the same as the configuration of the middleware part 20 according to Embodiment 1 shown in Fig. 2. The middleware part 20 according to Embodiment 3 is different from the middleware part 20 according to Embodiment 1 or 2 in arbitration information stored by an arbitration information storage part 22.

Concerning the selection phase and the combination phase, the arbitration information storage part 22 stores the priorities and the combination information in the same manner as in Embodiment 2. Concerning the transition phase, however, the arbitration information storage part 22 stores, as the arbitration information, status transition information which determines a primary request to be generated to correspond to the device status and a new inputted primary request.

If the arbitration information is status transition information, a request generating part 23 generates a new primary request to correspond to the device status and the new inputted primary requests, in accordance with the status transition

information. If no new primary request is inputted, the request generating part 23 generates a primary request that maintains the device status at that time.

[0046] Fig. 11 is a flowchart showing the operation of the middleware part 20 according to Embodiment 3.

S31 to S32 are the same as S21 to S22 shown in Fig. 8, and S34 to S37 are the same as S24 to S27 shown in Fig. 8.

In S33, if phase i is a selection phase, the request generating part 23 selects the primary request having the highest priority from among inputted primary requests. If phase i is a combination phase, the request generating part 23 generates a new primary request to correspond to the combination of the inputted primary requests, in accordance with the combination information. If phase i is a transition phase, the request generating part 23 generates a new primary request to correspond to the device operation status and the inputted primary requests, in accordance with the status transition information.

[0047] A case (a case in which the control apparatus 100 is built in a refrigerator) shown in Fig. 4 will be described.

Note that phase 1 is a transition phase and phase 2 is a selection phase. Arbitration information on phase 1 is arbitration information shown in Fig. 12. The arbitration information shown in Fig. 12 indicates new requests each to be generated to correspond to the operation status of the compressor and the inputted primary requests. For the sake of simplicity, a case where a weak/strong-protection function is active is omitted. Although not shown in Fig. 12, only a basic request is inputted when the refrigerator is started, and a normal operation is initiated.

[0048] In stage 1, only the basic request is inputted. Thus, the refrigerator operates at the 4th speed according to the inputted basic request, in the same manner as in

Embodiments 1 and 2.

[0049] <Stage 2>

After stage 1, the rapid-cooling button is depressed. Then a request input part 21 supplies, as input, the basic request outputted from a basic operation execution part 11 and the rapid-cooling request inputted from a function execution part 12A, to phase 1 (S31). Namely, a new rapid-cooling request is inputted to phase 1.

The request generating part 23 initializes the variable i to 1 (S32), and arbitrates the contention concerning phase 1 (S33). Since the operation status of the compressor is normal (4th speed) and the new rapid-cooling request is inputted, the request generating part 23 generates a rapid-cooling request in accordance with the arbitration information shown in Fig. 12. Since phase 1 is not the last phase (NO in S34), the request generating part 23 supplies, as input, the rapid-cooling request generated in S33 to phase 2 (S35). Then, the request generating part 23 adds 1 to the variable i to obtain 2, and puts the process back to S33 (S36).

The request generating part 23 arbitrates the contention concerning phase 2 (S33). Since there is no other primary request than the rapid-cooling request inputted from phase 1, a contention of the primary requests does not occur. Hence, the request generating part 23 selects the rapid-cooling request inputted from phase 1. Since phase 2 is the last phase (YES in S34), the request generating part 23 outputs the rapid-cooling request selected in S33 to a device control part 30 as the secondary request (S37).

In brief, at this point, the rapid-cooling request is outputted to the device control part 30, and the compressor operates at the 7th speed.

[0050] <Stage 3>

After that the ice-making button is depressed during the rapid-cooling

operation. Then, the request input part 21 supplies, as input, the basic request outputted from the basic operation execution part 11, the rapid-cooling request inputted from the function execution part 12A, and the ice-making request outputted from a function execution part 12B, to phase 1 (S31). That is, a new ice-making request is inputted to phase 1.

The request generating part 23 initializes the variable i to 1 (S32), and arbitrates the contention concerning phase 1 (S33). Since the operation status of the compressor is rapid-cooling (7th speed) and the ice-making request is inputted, the request generating part 23 generates a rapid-cooling & ice-making request in accordance with the arbitration information shown in Fig. 12. Since phase 1 is not the last phase (NO in S34), the request generating part 23 supplies, as input, the rapid-cooling & ice-making request generated in S43 to phase 2 (S35). Then, the request generating part 23 adds 1 to the variable i to obtain 2, and puts the process back to S33 (S36).

The request generating part 23 arbitrates the contention concerning phase 2 (S33). Since there is no other primary request than the rapid-cooling & ice-making request inputted from phase 1, a contention of the primary requests does not occur. Hence, the request generating part 23 selects the rapid-cooling & ice-making request inputted from phase 1. As phase 2 is the last phase (YES in S34), the request generating part 23 outputs the rapid-cooling & ice-making request selected in S33 to the device control part 30 as the secondary request (S37).

In brief, at this point, the rapid-cooling & ice-making request is outputted to the device control part 30, and the compressor operates at the 10th speed.

[0051] <Stage 4>

After that the weak-protection function is actuated during the rapid-cooling &

ice-making operation. Then, the request input part 21 supplies, as input, the basic request outputted from the basic operation execution part 11, the rapid-cooling request outputted from the function execution part 12A, and the ice-making request outputted from the function execution part 12B, to phase 1, and supplies, as input, the weak-protection function outputted from a function execution part 12C, to phase 2 (S31). Namely, a new weak-protection request is inputted to phase 2.

The request generating part 23 initializes the variable i to 1 (S32) and arbitrates the contention concerning phase 1 (S33). Since no new primary request is inputted, a rapid-cooling & ice-making request to maintain the current operation status of the compressor is generated. Since phase 1 is not the last phase (NO in S34), the request generating part 23 supplies, as input, the rapid-cooling & ice-making request generated in S33 to phase 2 (S35). Then, the request generating part 23 adds 1 to the variable i to obtain 2, and puts the process back to S33 (S26).

The request generating part 23 arbitrates the contention concerning phase 2 (S33). Since the rapid-cooling & ice-making request inputted from phase 1 and the weak-protection request outputted from the function execution part 12C are inputted, the request generating part 23 arbitrates the contention in accordance with the arbitration information shown in Fig. 6. As a result, the weak-protection request having a higher priority is selected. Since phase 2 is the last phase (YES in S34), the request generating part 23 outputs the weak-protection request selected in S33 to the device control part 30 as the secondary request (S37).

In brief, at this point, the weak-protection request is outputted to the device control part 30, and the compressor operates at the 2nd speed.

[0052] As described above, the control apparatus 100 according to Embodiment 3 stores the status transition information as the arbitration information, so that a new

primary request corresponding to the device status can be generated.

Hence, while simplifying the function of arbitrating the contending operation requests in the same manner as in Embodiments 1 and 2, control can be carried out more flexibly than in Embodiments 1 and 2.

[0053] In the above explanation, only the speed controlled by the primary request is referred to as the operation status of the compressor. However, information other than a parameter controlled by the primary request, for example, power per unit time consumed by the compressor, may be referred to as the operation status of the compressor. Also, the operation status of a device other than the compressor controlled by the primary request, for example, the opening of an expansion valve, may be referred to.

In brief, the status transition information in the transition phase may be information other than the parameter controlled by the primary request as the device operation status, or information on a device other than the device controlled by the primary request as the device operation status.

[0054] Embodiment 4.

In Embodiments 1 to 3, in each phase, one primary request is selected from among a plurality of inputted primary requests, or a new primary request is generated to correspond to the inputted primary requests and the like.

Embodiment 4 explains a case that includes a phase of correcting, by a primary request inputted from a function execution part 12, a primary request inputted from a previous phase.

[0055] Fig. 13 is a configuration diagram of a control apparatus 100 according to Embodiment 4.

The control apparatus 100 according to Embodiment 4 is different from the

control apparatus 100 according to Embodiments 1 to 3 in that each phase is determined to be a selection phase, a combination phase, a transition phase, or a correction phase.

The correction phase is a phase of correcting, by a primary request inputted from the function execution part 12, a primary request outputted from a basic operation execution part 11 or a primary request inputted from a previous phase, thereby generating a new primary request.

[0056] Fig. 14 is a configuration diagram of a middleware part 20 according to Embodiment 4.

The middleware part 20 according to Embodiment 4 is different from the middleware part 20 according to Embodiments 1 to 3 in that it is provided with a correction information storage part 24.

The correction information storage part 24 stores, for each correction phase, correction information indicating a method of correcting the primary request. The correction information indicates, for example, adding or multiplying a correction value expressed by a primary request inputted from the function execution part 12, to or by a primary request outputted from the basic operation execution part 11 or a primary request inputted from a previous phase.

Concerning the selection phase, the combination phase, and the transition phase, an arbitration information storage part 22 stores the priorities, the combination information, and the status transition information, in the same manner as in Embodiment 3. Concerning the correction phase, the arbitration information storage part 22 stores, as the arbitration information, the priorities, the combination information, or the status transition information. Note that the arbitration target is only the primary request inputted from the function execution part 12 and does not include the primary request outputted from the basic operation execution part 11 or a primary request

inputted from the previous phase.

In the correction phase, in accordance with the arbitration information, a request generating part 23 generates a primary request being the result of arbitration of the primary requests inputted from the function execution part 12. Then, in accordance with the correction information, the request generating part 23 corrects the primary request outputted from the basic operation execution part 11 or the primary request inputted from a previous phase, by a correction value expressed by the generated primary request, thereby generating a new primary request.

[0057] Fig. 15 is a flowchart showing the operation of the middleware part 20 according to Embodiment 4.

S41 to S42 are the same as S31 to S32 shown in Fig. 11, and S44 to S47 are the same as S34 to S37 shown in Fig. 11.

In S43, if phase i is a selection phase, the request generating part 23 selects the primary request having the highest priority from among inputted primary requests. If phase i is a combination phase, the request generating part 23 generates a new primary request to correspond to the combination of the inputted primary requests, in accordance with the combination information. If phase i is a transition phase, the request generating part 23 generates a new primary request to correspond to the device operation status and the inputted primary request, in accordance with the status transition information. If phase i is a correction phase, the request generating part 23 corrects the primary request outputted from the basic operation execution part 11 or the primary request inputted from a previous phase, by the primary request inputted from the function execution part 12, thereby generating a new primary request.

[0058] A case (a case in which the control apparatus 100 is built in a refrigerator) shown in Fig. 4 will be described.

Note that phase 1 is a correction phase and phase 2 is a selection phase. Arbitration information on phase 1 is the arbitration information shown in Fig. 16. The arbitration information shown in Fig. 16 indicates the priorities of a rapid-cooling request and an ice-making request. According to Fig. 16, the rapid-cooling request has a lower priority and the ice-making request has a higher priority.

Also, the correction information storage part 24 stores, as the correction information on phase 1, adding a correction value to the basic request.

A basic request as the primary request outputted from the basic operation execution part 11 requests operation at the 4th speed. A weak-protection request requests operation at the 2nd speed. A strong-protection request requests operation at the 0th speed. The rapid-cooling request indicates 3rd speed as the correction value. The ice-making request indicates 1st speed as the correction value.

[0059] In stage 1, only the basic request is inputted. Thus, the refrigerator operates at the 4th speed according to the inputted basic request, in the same manner as in Embodiments 1 to 3.

[0060] <Stage 2>

After stage 1, the rapid-cooling button is depressed. Then, a request input part 21 supplies, as input, the basic request outputted from the basic operation execution part 11 and the rapid-cooling request inputted from a function execution part 12A, to phase 1 (S41).

The request generating part 23 initializes a variable i to 1 (S42), and arbitrates the contention concerning phase 1 (S43). Since the rapid-cooling request is inputted in addition to the basic request, the request generating part 23 selects the rapid-cooling request. Then, in accordance with the correction information, the request generating part 23 adds a correction value of 3rd speed expressed by the selected rapid-cooling

request to the 4th speed expressed by the basic request, thereby generating a rapid-cooling request requesting operation at the 7th speed. Since phase 1 is not the last phase (NO in S44), the request generating part 23 supplies, as input, the rapid-cooling request selected in S43, to phase 2 (S45). Then, the request generating part 23 adds 1 to the variable i to obtain 2, and puts the process back to S43 (S46).

The request generating part 23 arbitrates the contention concerning phase 2 (S43). Since there is no other primary request than the rapid-cooling request inputted from phase 1, a contention of the primary requests does not occur. Hence, the request generating part 23 selects the rapid-cooling request inputted from phase 1. Since phase 2 is the last phase (YES in S44), the request generating part 23 outputs the rapid-cooling request selected in S43 to a device control part 30 as the secondary request (S47).

In brief, at this point, the rapid-cooling request is outputted to the device control part 30, and the compressor operates at the 7th speed.

[0061] <Stage 3>

After that the ice-making button is depressed during the rapid-cooling operation. Then, the request input part 21 supplies, as input, the basic request outputted from the basic operation execution part 11, the rapid-cooling request outputted from the function execution part 12A, and the ice-making request outputted from a function execution part 12B, to phase 1 (S41).

The request generating part 23 initializes the variable i to 1 (S42), and arbitrates the contention concerning phase 1 (S43). Since the rapid-cooling request and the ice-making request are inputted in addition to the basic request, the request generating part 23 arbitrates the contention in accordance with the arbitration information shown in Fig. 16. As a result, the ice-making request having a higher

priority is selected. In accordance with the correction information, the request generating part 23 adds a correction value of the 1st speed expressed by the selected ice-making request to the 4th speed expressed by the basic request, thereby generating an ice-making request requesting operation at the 5th speed. Since phase 1 is not the last phase (NO in S44), the request generating part 23 supplies, as input, the ice-making request selected in S43, to phase 2 (S45). Then, the request generating part 23 adds 1 to the variable i to obtain 2, and puts the process back to S43 (S46).

The request generating part 23 arbitrates the contention concerning phase 2 (S43). Since there is no other primary request than the ice-making request inputted from phase 1, a contention of the primary requests does not occur. Hence, the request generating part 23 selects the ice-making request inputted from phase 1. Since phase 2 is the last phase (YES in S44), the request generating part 23 outputs the ice-making request selected in S43 to the device control part 30 as the secondary request (S47).

In brief, at this point, the ice-making request is outputted to the device control part 30, and the compressor operates at the 5th speed.

[0062] <Stage 4>

After that the weak-protection function is actuated during the ice-making operation. Then, the request input part 21 supplies, as input, the basic request outputted from the basic operation execution part 11, the rapid-cooling request outputted from the function execution part 12A, and the ice-making request outputted from the function execution part 12B, to phase 1, and supplies, as input, the weak-protection function inputted from a function execution part 12C, to phase 2 (S41).

The request generating part 23 initializes the variable i to 1 (S42), arbitrates the contention concerning phase 1, in the same manner as in stage 3, and supplies, as input, the ice-making request to phase 2 (S43 to S45). Then, the request generating part 23

adds 1 to the variable i to obtain 2, and puts the process back to S3 (S46).

The request generating part 23 arbitrates the contention concerning phase 2 (S43). Since the ice-making request inputted from phase 1 and the weak-protection request outputted from the function execution part 12C are inputted, the request generating part 23 arbitrates the contention in accordance with the arbitration information shown in Fig. 6. As a result, the weak-protection request having a higher priority is selected. Since phase 2 is the last phase (YES in S44), the request generating part 23 outputs the weak-protection request selected in S43 to the device control part 30 as the secondary request (S47).

In brief, at this point, the weak-protection request is outputted to the device control part 30, and the compressor operates at the 2nd speed.

[0063] As described above, the control apparatus 100 according to Embodiment 4 stores the new correction information so as to correct the primary request generated in the previous phase, thereby generating a new primary request.

Hence, while highly simplifying the function of arbitrating the contending operation requests in the same manner as in Embodiments 1 to 3, control can be carried out more flexibly.

[0064] In the above explanation, if an additional function is not active, the operation speed of the compressor is always 4th speed. However, even if the additional function is not active, the operation speed of the compressor may be altered depending on the ambient temperature or the like of the refrigerator, for the purpose of keeping, for example, a constant refrigerator temperature. In brief, the basic operation execution part 11 may output a basic request requesting different operation speeds depending on the ambient temperature or the like of the refrigerator.

In that case, as described in Embodiment 4, when the additional function such

as the rapid-cooling function and the ice-making function are actuated, if the current operation speed is corrected by a correction value corresponding to the additional function, more appropriate control can be performed.

Meanwhile, there is an additional function which, like a protection function, must be controlled to a predetermined speed regardless of the current operation speed. In that case, if control is carried out at an operation speed specified by the additional function regardless of the current operation speed, appropriate control is enabled, as described in Embodiment 1.

[0065] Embodiment 5.

Embodiments 1 to 4 describe the process of the control apparatus 100 by referring to a case wherein the control apparatus 100 is built in the refrigerator. The control apparatus 100 may, however, be built in consumer electronics other than a refrigerator, or another device.

Embodiment 5 explains a case in which the control apparatus 100 is built in another device.

[0066] Fig. 17 shows a case in which the control apparatus 100 is built in an air-conditioner.

In general, the air-conditioner is provided with a compressor (an example of the device). The air-conditioner can control the room temperature by changing the operation speed (rotation speed rpm or rotation frequency Hz) of the compressor. A case will be described in which the control apparatus 100 controls the operation speed of the compressor of the air-conditioner in the cooling operation.

[0067] When the air-conditioner is turned on, it carries out a basic operation (basic operation execution part 11) of cooling the room continuously.

The air-conditioner is provided with various other functions (function

execution part 12). In this example, the air-conditioner is provided with an additional function 1 (high-power operation), an additional function 2 (arbitrary frequency control), a protection function 1 (discharge temperature overrise prevention), a protection function 2 (condensation temperature overrise prevention), a protection function 3 (MIN frequency protection), a protection function 4 (countermeasure for vibration in stopping), and a protection function 5 (discharge overrise stop).

High-power operation (additional function 1) is a function actuated when the user operates an operation button in order to turn up the air-conditioning performance. The additional function 1 adds a predetermined correction value to the operation speed of the compressor, thereby increasing the operation speed of the compressor.

Arbitrary frequency control (additional function 2) is a function employed in inspection in factory shipment and the like. The additional function 2 sets a predetermined operation speed of the compressor and actuates the compressor.

Discharge temperature overrise prevention (protection function 1) is a function that prevents the compressor from being broken by overrise of the temperature or pressure of the discharge gas. When a sensor detects an abnormal rise of the temperature or pressure, the protection function 1 subtracts a predetermined correction value from the operation speed of the compressor, thereby decreasing the operation speed of the compressor.

Condensation temperature overrise prevention (protection function 2) is a function that suppresses pressure rise of the high-pressure side so that the refrigerant system (compressor, heat exchanger) will not be broken. When a sensor detects an abnormal rise of the pressure, the protection function 2 subtracts a predetermined correction value from the operation speed of the compressor, thereby decreasing the operation speed of the compressor.

MIN frequency protection (protection function 3) is a function that maintains the minimum operation speed so as to ensure the refrigerant circulation amount. This is because if the operation speed decreases more than necessary, it imposes a load to the compressor. When the operation speed falls below a predetermined speed, the protection function 3 sets the operation speed of the compressor to the minimum speed.

Countermeasure for vibration in stopping (protection function 4) is a function that prevents the piping from being disconnected by vibration occurring when stopping the compressor. The protection function 3, before the compressor is stopped, sets the operation speed of the compressor to a predetermined speed for a predetermined period of time.

Discharge overrise stop (protection function 5) is a function that brings the compressor to emergency stop when a sensor detects an abnormal rise in the temperature or pressure of the discharge gas. The protection function 5 sets the operation speed of the compressor to 0.

[0068] Primary requests outputted from the basic operation and the additional function 1 are inputted to phase 1 (correction phase). A primary request outputted from the additional function 2 is inputted to phase 2 (selection phase). Primary requests outputted from the protection functions 1 and 2 are inputted to phase 3 (correction phase). Primary requests outputted from the protection functions 3, 4, and 5 are inputted to phase 4 (selection phase).

In phase 3, as arbitrary information, combination information is set according to which: if only the protection function 1 is active, the correction value of the protection function is selected; if only the protection function 2 is active, the correction value of the protection function 2 is selected; and if both the protection functions 1 and 2 are active, both of the two correction values are selected.

[0069] Fig. 18 is a chart showing secondary requests outputted in the example shown in Fig. 17.

In phase 1, the operation is premised on a speed F0 of the compressor which is set for the basic operation. If the additional function 1 is active, a speed ΔF1 is added to the speed F0. Thus, the speed F0 or a speed F0+ΔF1 is outputted from phase 1.

In phase 2, either the speed (the speed F0 or the speed F0+ΔF1) outputted in phase 1 or a speed F2 outputted from the additional function 2 is outputted. Thus, the speed F0, the speed F0+ΔF1, or the speed F2 is outputted from phase 2.

In phase 3, the speed outputted from phase 2 is corrected by a speed -ΔF3A if only the protection function 1 is active; is corrected by a speed -ΔF3B if only the protection function 2 is active; and is corrected by both the speed -ΔF3A and the speed -ΔF3B if both the protection functions 1 and 2 are active. As a result, one of 9 different speeds corresponding to the combination is outputted.

In phase 4, one of the speed outputted in phase 3 and the speeds outputted from the protection functions 3, 4 and 5 is outputted.

Therefore, a total of 15 different speeds are outputted depending on the active statuses of the additional function and the protection function.

[0070] Fig. 19 shows a case in which a general control apparatus is built in an air-conditioner.

In the general control device, each function determines whether or not to output an operation request to the device driver by checking the active statuses of other functions in the control device. Each function accordingly includes very complicated condition determinations, which leads to a low maintainability, a low reusability, and a low expandability.

In contrast, as shown in Fig. 17, when the control apparatus 100 is used, the

condition determinations as to whether or not to output an operation request to the device driver are well-organized, thus providing a high maintainability, a high reusability, and a high expandability.

[0071] Fig. 20 is an explanatory diagram of a DC (direct current) motor-driven automated cart.

The automated cart is provided with a motor (an example of the device). The operation speed of the motor is changed by changing the voltage to be applied to the motor, so that the moving speed of the automated cart can be controlled.

The automated cart transports a load by simple linear displacements. The automated cart is provided with a torque sensor, a speed sensor, and a temperature sensor which accompany the motor, and a forward obstacle sensor on the front side in the traveling direction and a backward obstacle sensor on the rear side in the traveling direction.

[0072] Fig. 21 shows a case in which the control apparatus 100 is built in the automated cart shown in Fig. 20.

An example will be described in which the control apparatus 100 controls the operation speed of the motor of the automated cart.

[0073] When the automated cart is turned on, it performs a basic operation (basic operation execution part 11) of transporting the load by simple linear displacements. In the basic operation, different constant voltage values for operation start, constant-speed driving, and stopping are outputted to the motor.

The automated cart is also provided with various other functions (function execution part 12). In this example, the automated cart is provided with an additional function 1 (speed keeping), an additional function 2 (energy-saving operation), an additional function 3 (high-speed operation), an additional function 4 (safe operation), a

protection function 1 (backward-obstacle acceleration), a protection function 2 (overload deceleration), a protection function 3 (forward-obstacle stop), and a protection function 4 (emergency stop).

The additional function 1 (speed keeping) is a function of keeping the speed during constant-speed traveling. In accordance with the information from the speed sensor, the additional function 1 corrects the voltage value and to adjust the speed.

The additional function 2 (energy-saving operation) is a function of realizing an energy-saving operation. The additional function 2 subtracts a correction value to lower the voltage value.

The additional function 3 (high-speed operation) is a function of driving at a high speed so that, for example, efficient transport is achieved. The additional function 3 outputs the maximum voltage to accelerate the automated cart.

The additional function 4 (safe operation) is a function of operating the automated cart at a low speed, in a case where there is a danger of load collapse, to achieve safe driving. The additional function 4 outputs a low voltage to decelerate the automated cart.

The protection function 1 (backward-obstacle acceleration) is a function of preventing a risk in a case where the backward-obstacle sensor detects a backward obstacle. Upon detection of the backward obstacle, the protection function 1 adds a correction value to the current voltage value to accelerate the automated cart.

The protection function 2 (overload deceleration) is a function of detecting an overload state by a torque sensor, thereby preventing the motor from being broken. Upon detection of the overload state, the protection function 2 subtracts a correction value from the current voltage value to decelerate the automated cart.

The protection function 3 (forward-obstacle stop) is a function of preventing a

collision in a case where the forward-obstacle sensor detects a forward obstacle. Upon detection of the forward obstacle, the protection function 3 outputs a voltage value 0 to apply a quick brake.

The protection function 4 (emergency stop) is a function of preventing danger to the motor in a case where the temperature sensor detects an abnormal temperature rise of the motor. Upon detection of the abnormal temperature rise of the motor, the protection function 4 outputs a voltage value 0 to bring the motor to emergency stop.

[0074] Primary requests outputted from the basic operation and the additional functions 1 and 2 are inputted to phase 1 (correction phase). Primary requests outputted from the additional functions 3 and 4 are inputted to phase 2 (selection phase). Primary requests outputted from the protection functions 1 and 2 are inputted to phase 3 (correction phase). Primary requests outputted from the protection functions 3 and 4 are inputted to phase 4 (selection phase).

In phase 1, if the additional functions 1 and 2 are active simultaneously, the arbitration information is set such that correction is done with both of the two correction values.

[0075] In phase 1, the operation is premised on a voltage value V0 of the motor which is set for the basic operation. If only the additional function 1 is active, correction is done with a voltage value $\pm\Delta V1A$. If only the additional function 2 is active, correction is done with a voltage value $-\Delta V1A$. If both of the additional functions 1 and 2 are active, correction is done with both of the voltage value $\pm\Delta V1A$ and the voltage value $-\Delta V1A$.

In phase 2, one of the voltage value outputted from phase 1 and the voltage values outputted from the additional functions 3 and 4 is selected.

In phase 3, the voltage value outputted from phase 2 is corrected with either

one of the voltage values outputted from the protection functions 1 and 2.

In phase 4, either the voltage value outputted from phase 3 or one of the voltage values outputted from the protection functions 3 and 4 is selected.

[0076] As described, the control apparatus 100 according to Embodiments 1 to 4 can be built not only in a refrigerator but also in any product in which various types of devices such as an actuator are controlled by operation requests outputted from various functions using a microcomputer. When the control apparatus 100 according to Embodiments 1 to 4 is employed, the function of arbitrating contending operation requests can have a very simple configuration. As a result, maintenance and function expansion of the product can be performed easily.

[0077] The control apparatus 100 according to Embodiments 1 to 4 can be mounted in, for example, a robot, a manipulator, an automobile, and artificial satellites in which many drive motors are mounted, in addition to in the home electronics and automated cart described above.

[0078] In the above description, the control value such as the operation speed of the compressor is outputted as the secondary request from the middleware part 20 to the device control part 30.

However, not only the control value but also information indicating various types of processes such as a normal process, an initialization process, an emergency stop process, and a demonstration mode process may be outputted as the secondary request from the middleware part 20 to the device control part 30. In that case, the device control part 30 outputs a control parameter corresponding to the process indicated by the secondary request, to the device driver 40, to cause the device to execute the process.

[0079] In the above description, the device control part 30 outputs the secondary

request outputted from the middleware part 20, to the device driver 40 merely as a parameter, thereby controlling the device. For example, regarding the refrigerator described above, the device control part 30 outputs a speed indicated by the secondary request to the device driver 40 as the parameter, to cause the compressor to operate at that speed.

However, in addition to outputting the secondary request to the device driver 40 merely as the parameter, the device control part 30 may, for example, also calculate, from the secondary request, parameters that control other devices. For example, in the case of the refrigerator described above, in addition to outputting the speed indicated by the secondary request as the parameter to the device driver 40 of the compressor, the device control part 30 may also calculate the rotation speed of a fan that corresponds to the speed indicated by the secondary request, and may output the calculated rotation speed, as the parameter to a device driver 40 of the fan. Furthermore, the device control part 30 may output a parameter that turns on a lamp corresponding to the speed indicated by the secondary request, to a device driver 40 of the lamp.

[0080] The device control part 30 stores the limit value (upper-limit value and lower-limit value) of the parameter. If a value exceeding the limit value is inputted as the secondary request, the device control part 30 may supply the limit value to the device driver 40 as a control parameter. In brief, the device control part 30 may also serve as the final protection function.

[0081] In the middleware part 20, it is necessary to determine the phases to which the primary requests outputted from the respective function execution parts 12 are to be inputted, and the arbitration information in the respective phases for each device to be controlled.

Note that some devices can be controlled under the common conditions. For

example, in the case of the air-conditioner described above, the speed control of the fan motor that supplies air into the room can be realized by control that is the same as the control for the compressor. In this manner, regarding devices that can be controlled under the common conditions, the primary requests can be inputted to the same phase, and the devices can be controlled using the same arbitration information. This can reduce the storage area of the storage device which stores the arbitration information.

[0082] When a primary request outputted from a function having a high priority is inputted to a former or latter phase, the higher the priority of the function that outputted the primary request, the more likely the primary request is selected by the function of that phase of arbitrating the contending operation requests.

[0083] Fig. 22 shows an example of the hardware configuration of the control apparatus 100 according to the above embodiments.

As shown in Fig. 22, the control apparatus 100 includes a CPU 911 (Central Processing Unit; also referred to as central processing device, processing device, computation device, microprocessor, microcomputer, or processor) which executes programs. The CPU 911 is connected to a ROM 913, a RAM 914, and a communication board 915 via a bus 912, and controls these hardware devices.

[0084] The ROM 913 is an example of a nonvolatile memory. The RAM 914 is an example of a volatile memory. The ROM 913 and the RAM 914 are examples of the storage device (memory). The communication board 915 is an example of a communication device.

[0085] The ROM 913 stores programs 916 and files 917. The CPU 911 executes each program of the programs 916.

[0086] The programs 916 store software and programs that execute the functions described as the "application part 10", "middleware part 20", "device control part 30",

"device driver 40", "basic operation execution part 11", "function execution part 12", "request input part 21", "request generating part 23", and the like in the above description. The programs 916 store other programs as well. The programs are read and executed by the CPU 911.

The files 917 store information such as the "operation request", "arbitration information", "correction information", and the like of the above explanation. The "file" and "database" are stored in a recording medium such as a disk or memory. The information, data, signal values, variable values, and parameters stored in the recording medium such as the disk or memory are read out to the main memory or cache memory by the CPU 911 through a read/write circuit, and are used for the operations of the CPU 911 such as extraction, search, look-up, comparison, computation, calculation, process, output, print, and display. The information, data, signal values, variable values, and parameters are temporarily stored in the main memory, cache memory, or buffer memory during the operations of the CPU 911 including extraction, search, look-up, comparison, computation, calculation, process, output, print, and display.

[0087] The arrows of the flowcharts in the above explanation mainly indicate input/output of data and signals. The data and signal values are recorded in the memory of the RAM 914, in the recording medium such as an optical disk, or in an IC chip. The data and signals are transmitted online via a transmission medium such as the bus 912, signal lines, or cables; or electric waves.

The "part" in the above explanation may be a "circuit", "device", "equipment", "means" or "function"; or a "step", "procedure", or "process". The "device" may be a "circuit", "equipment", "means", or "function"; or a "step", "procedure", or "process". The "process" may be a "step". Namely, the "part" may be implemented as firmware stored in the ROM 913. Alternatively, the "part" may be implemented by only

software; by only hardware such as an element, a device, a substrate, or a wiring line; by a combination of software and hardware; or furthermore by a combination of software, hardware, and firmware. The firmware and software are stored, as a program, in the recording medium such as the ROM 913. The program is read by the CPU 911 and executed by the CPU 911. Namely, the program causes the computer or the like to function as a "part" described above. Alternatively, the program causes the computer or the like to execute the procedure and method of the "part" described above.

Reference Signs List

[0088] 10: application part; 11: basic operation execution part; 12: function execution part; 20: middleware part; 21: request input part; 22: arbitration information storage part; 23: request generating part; 24: correction information storage part; 30: device control part; 40: device driver; 100: control apparatus

Claims

[Claim 1] A control apparatus which arbitrates contending operation requests outputted from a plurality of APPs, APP standing for application program, and controls a device, comprising:

a request input part which supplies, as input, the operation requests outputted from the APPs, each to a phase predetermined for a corresponding APP among a plurality of phases arranged in a fixed order;

an arbitration information storage part which stores, for each phase, arbitration information;

a request generating part which generates an operation request, being a result of arbitrating a contention of the operation requests inputted, in the phases sequentially starting with the first phase and in accordance with the arbitration information stored in the arbitration information storage part, and supplies, as input, the operation request generated, to a next phase; and

a device control part which controls an operation of the device in accordance with the operation request generated by the request generating part in the last phase.

[Claim 2] The control apparatus according to claim 1,

wherein the arbitration information storage part, concerning a selection phase among the plurality of phases, stores a priority concerning the operation request inputted, as the arbitration information, and

wherein the request generating part, when in the selection phase, selects a single operation request having a high priority from among the operation requests inputted, and supplies, as input, the operation request selected, to a next phase.

[Claim 3] The control apparatus according to claim 1 or 2,

wherein the arbitration information storage part, for a combination phase

among the plurality of phases, stores combination information that determines an operation request to be generated, as the arbitration information in accordance with a combination of the operation requests inputted, and

wherein the request generating part, when in the combination phase, generates a new operation request from the operation requests inputted, to correspond to the combination information, and supplies, as input, the operation request generated, to a next phase.

[Claim 4] The control apparatus according to any one of claims 1 to 3,

wherein the arbitration information storage part, for a transition phase among the plurality of phases, stores status transition information that determines an operation request to be generated, as the arbitration information to correspond to a combination of an operation status of the device and the operation requests inputted, and

wherein the request generating part, when in the transition phase, generates a new operation request from the operation requests inputted, in accordance with the status transition information, and supplies, as input, the operation request generated, to a next phase.

[Claim 5] The control apparatus according to any one of claims 1 to 4,

wherein the arbitration information storage part, for a correction phase among the plurality of phases, stores a priority concerning the operation requests outputted from the APPs, as the arbitration information, and

wherein the request generating part, when in the correction phase, selects a single operation request having a high priority from among the operation requests outputted from the APPs, corrects the operation request generated in a previous phase, with the operation request selected, and supplies, as input, the operation request corrected, to a next phase.

[Claim 6] The control apparatus according to any one of claims 1 to 5, wherein the request input part supplies, as input, an operation request having a higher priority to an earlier or later phase.

[Claim 7] A control method of arbitrating contending operation requests outputted from a plurality of APPs, APP standing for application program, and controlling a device, comprising:

a request input step of, with a processing device, supplying, as input, the operation requests outputted from the APPs, each to a phase predetermined for a corresponding APP among a plurality of phases arranged in a fixed order;

a request generating step of, with the processing device, generating an operation request, being a result of arbitrating a contention of the operation requests inputted, in the phases sequentially starting with the first phase and in accordance with arbitration information stored in a storage device in advance for each phase, and supplying, as input, the operation request generated, to a next phase; and

a device control step of, with the processing device, controlling an operation of the device in accordance with the operation request generated in the request generating step in the last phase.

[Claim 8] A control program which arbitrates contending operation requests outputted from a plurality of APPs, APP standing for application program, and controls a device, comprising:

a request input process of supplying, as input, the operation requests outputted from the APPs, each to a phase predetermined for a corresponding APP among a plurality of phases arranged in a fixed order;

a request generating process of generating an operation request, being a result of arbitrating a contention of the operation requests inputted, in the phases sequentially

starting with the first phase and in accordance with arbitration information stored in a storage device in advance for each phase, and supplying, as input, the operation request generated, to a next phase; and

a device control process of controlling an operation of the device in accordance with the operation request generated in the request generating process in the last phase.