



(51) International Patent Classification:

G06Q 10/04 (2012.01) G06Q 50/06 (2012.01)

G05D 23/19 (2006.01)

(21) International Application Number:

PCT/EP2019/065002

(22) International Filing Date:

07 June 2019 (07.06.2019)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

19152612.8 18 January 2019 (18.01.2019) EP

(71) Applicant: **ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE (EPFL)** [CH/CH]; EPFL-TTO, EPFL Innovation Park J, 1015 Lausanne (CH).

(72) Inventor: **MALAMUD, Semyon**; Ch. de la Fauvette 10A, 1012 LAUSANNE (CH).

(74) Agent: **BITTNER, Peter** et al.; Herrenwiesenweg 2, 69207 Sandhausen (DE).

(81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with declaration under Article 17(2)(a); without abstract; title not checked by the International Searching Authority

(54) Title: SYSTEM AND METHOD FOR USER DATA INPUT TO MODIFY CONFIGURATIONS OF PHYSICAL ENTITIES

(57) Abstract:



WO 2020/147986 A2

System and Method for User Data Input to Modify Configurations of Physical Entities

Technical Field

[0001] The present invention generally relates to electronic data processing, and more particularly, relates to methods, computer program products and systems for improving data input by a user to modify configuration settings for physical entities. Thereby, a physical entity is a real-world object which has physical existence.

Background

[0002] In many real-world situations, human users are confronted with the technical task to select configurations of physical entities so that the physical entity can be transformed into an optimal state. Typically, a configurable physical entity is in an initial (or default) state but can transition into other states in accordance with respective configurations. However, for a human user it is difficult to select an optimized configuration under certain constraints unless the user exactly understands the behavior of the physical entity. A property of many physical entities is that linkages exist between the different states of the entity. In many situations, a single action (such as a choice of an input influencing the future behavior of the physical entity) influences the behavior of the physical entity in multiple future states. Such linkages between the states of the physical entity can be quite complex, so that it becomes an impossible task for a human being to grasp the joint impact of a change in any given input on the plurality of future states.

[0003] An example of a physical entity is an electrical device (e.g., a robot or machine) which uses energy with a constraint that the use of a particular energy type is to be minimized. Another example of a real-world device is a vehicle (e.g., car or aircraft) powered by a fuel driven engine with a constraint that fuel consumption is to be optimized. Another example of a physical entity is a heating system of a (smart) building with a constraint to keep the room temperature within a narrow range of a target temperature value. Another example of a physical entity is a data structure stored on a storage medium, with the objective that this data structure needs to be processed as fast as possible, but at the same time not limiting too much the respective computing resources (such as GPU, CPU, and memory) which

should also be available to users for other objectives. Further examples are given in the description below.

Summary

[0004] There is a need to provide systems and methods enabling a robust user interaction with a data input device supporting the user to select an optimized configuration for the physical entity under given constraints.

[0005] A physical entity is characterized by two features:

(1) a configuration (a collection of state-value pairs for multiple values of so-called base parameters), and

(2) a set of inputs consistent with the given configuration and satisfying admissibility (inequality) constraints. This set of inputs is characterized as a multi-dimensional semi-analytic set, defined by equalities and inequalities involving real analytic functions. The definition of semi-analytic can be obtained at the website

<http://mathworld.wolfram.com/Semianalytic.html>: $X \subseteq \mathbb{R}^n$ is semi-analytic if, for all $x \in \mathbb{R}^n$ there is an open neighborhood $X \cap U$ such that $X \cap U$ is a finite Boolean combination of sets $\{\bar{x} \in U : f(\bar{x}) = 0\}$ and $\{\bar{x} \in U : g(\bar{x}) > 0\}$, where $f, g : U \rightarrow \mathbb{R}$ are analytic.

Admissibility (inequality) constraints are either defined by the user (who defines which inputs are considered as being admissible), or they are defined by the physical constraints on the input combinations, as determined by the real world conditions.

[0006] The physical entity exhibits linkages across states if, for at least one of the inputs, a modification of this input affects the state of the configuration for more than one value of a base parameter. Thereby, configurations are designed to take into account the linkages between the different states of the physical entity. In other words, a change in the input for a particular configuration may affect a plurality of further potential states of the physical entity because of dependencies (linkages) which exist between the potential states. Robust user interaction, as used herein, refers to a user interaction with the computer system which reduces the likelihood for user inputs which lead to a poor or sub-optimal configuration of the physical entity that does not efficiently account for the linkages between different states of the physical entity.

[0007] This problem is solved by the features of the independent claims. In one embodiment, a computer-implemented method is provided for editing a configuration of a physical entity to optimize a future state of the physical entity. The claimed method supports the user in selecting a configuration that is perceived to be optimal by the user. A human user is not able to grasp the large (and sometimes very large) number of possible input combinations that are consistent with the selected configuration. It is assumed as a prerequisite that for any given configuration, the set of input combinations consistent with this configuration is given by a semi-analytic set. The nature of this set depends on the linkages between different states of the physical entity. The method takes these linkages into account by explicitly using the system of equations and inequalities that characterize a semi-analytic set of input combinations that are consistent with a given configuration and satisfy the admissibility constraints. Once the system of equations is formulated, any optimization algorithm (such as gradient descent methods, linear and non-linear programming, static and dynamic programming, Bellman equations) can be used to optimize over this semi-analytic set using explicitly the derived characterization in terms of equations and inequalities.

[0008] Examples of physical entities include but are not limited to electrical devices, heater devices, vehicles, data structures for storing data records in data storage devices, etc. Various examples of scenarios related to different physical entities are discussed in the detailed description. For each application scenario, the corresponding semi-analytic set of input combinations that are consistent with a given configuration is explicitly characterized. This characterization serves as an input to any optimization algorithm chosen to select the optimal input combination.

[0009] The configuration includes a plurality of potential future states for the physical entity. Such potential future states are referred to as “potential states” herein. Each configuration is described as a function of a base parameter on which the potential states depend. It is assumed that the base parameter takes values in an interval of real numbers, defined by the physical nature of the physical entity. To reduce the set of such functions to a number which can be processed by the computer system in a reasonable amount of time (to allow for a real time response to the user via the GUI), a discrete grid of base parameter values is selected and each configuration is described as a piece-wise linear function that is uniquely defined

by the discrete values of the function on the discrete base parameter grid. This grid may be chosen by the underlying algorithm as an approximation. Or, it may be that the nature of input combinations is such that all physically possible configurations are defined by piece-wise linear functions on a given grid. Since a piece-wise linear function on a grid is uniquely defined by its values on this grid, in the following a configuration is identified by its values on the discrete grid. It is to be noted that each state is defined by a discrete value pair with a state value and a base parameter value. In other words, the configuration describes a discrete number of different states of the physical entity which can prevail at a pre-specified future time point or at different future time points. The user is now confronted with the task to modify the configuration in such a way that the future state of the physical entity gets optimized under a particular constraint. Typically, such constraints are user-defined or pre-defined. Thereby, it may be critical to take into account linkages which may exist between different future states of the physical entity. The constraints are always given in terms of semi-analytic equalities and inequalities defining the set of admissible input combinations as well as the set of admissible configurations that can actually be implemented in the real world given the physical constraints on the physical entity. In one embodiment, these admissibility (inequality) constraints can be verified by a consistency check module of the computer system. A configuration is executable under real-world conditions if and only if there exists a non-empty set of input combinations consistent with this configuration and satisfying the inequality constraints.

[0010] Initially, the user selects the base parameter and the type of potential states for which an optimized configuration is to be determined. The selection of the user is received by the system via appropriate input means of the user interface (e.g., mouse, touchscreen, audio, etc.). For example, the user interface may propose a list of potential value pairs with each pair including a particular base parameter with a respective state type for which configuration optimizations are supported by an optimizer component of a computer system which is communicatively coupled with the graphical user interface component used by the user. Alternatively, the graphical user interface may provide the user with a questionnaire where the answers of the user finally identify the optimization objective of the user, thus leading implicitly to the selection of base parameter and state type for the further user input process.

[0011] Once the optimization objective of the user is received in the form of the base parameter and the state type, the graphical user interface provides an initial visual representation of a default configuration. The default configuration includes a predefined list of configuration value pairs, each configuration value pair with a base parameter value and a corresponding potential state value. Such value pairs may be stored in a data structure or they may be entered by the user. That is, a configuration value pair unambiguously defines a potential state of the physical entity.

[0012] In the following sections the example of an electrical device (e.g., a heating control device of a smart house) as a physical entity is used. However, this example is used for explanatory purposes only with no intention to limit the scope of the claims. A person skilled in the art is able to transfer the teachings to any other appropriate physical entities with respective base parameter and state type. According to the above definition of a physical entity via the two characteristics described earlier, in this example the goal is to define a configuration and a set of input combinations that is consistent with the given configuration. For example, one may define a default configuration for the electrical device representing the interior temperature of the house as a function of the out-door temperature. For example, input combinations can be defined as consumption of non-renewable energy (e.g., gas, oil, wood) or renewable energy (e.g., generated by photovoltaic cells). The user intends to optimize this configuration under one or more predefined constraints. Thereby, the objective function for the user does not depend on the configuration itself but on the chosen input combinations that are consistent with the configuration. The constraints on the underlying input combinations may be defined by the user or may be predefined for the optimizer component. In the electrical device example the constraint may be that the operational output of the electrical device should be optimized by taking into account all possible future values of the base parameter, but at the same time, the use of non-renewable energy by the device should be minimized. Thereby, all possible future values of the base parameter correspond to the base parameter values associated with all the potential states of the respective configuration. Additional inequality constraints on the admissible input combinations may come from the physical nature of the physical entity. For example, using energy from photovoltaic cells is only possible when there is sufficient solar light intensity.

[0013] The plurality of the potential states of the physical entity are represented by interactive graphical user interface objects which correspond to default function values at respective base parameter values (i.e., potential future base parameter values). In other words, each configuration value pair is represented by a respective interactive graphical user interface object in a visual representation of the configuration on the graphical user interface. An interactive user interface object, as used herein, is an object which is responsive to a user interaction. For example, the object may be represented as an icon which can be moved by the user by a drag action. Such drag action may be performed via a respective gesture applied to the icon on a touch screen, or it may be performed by selecting the icon (e.g., with a mouse click) and using user interaction means (e.g., a slider) to move the icon. For example, the visual representation of the default configuration may be in the form of a graphical chart showing the potential state values as a function of the base parameter. Thereby, predefined configuration value pairs are connected through interpolation which is fit into the configuration value pairs for the base parameter on the grid. The values of this curve for intermediate base parameter values (i.e., for base parameter values between the discrete base parameter values on the grid) represent state values for the intermediate base parameter values. An interactive user interface object representing a potential configuration value pair allows the user to modify the values of this value pair by applying the drag action. Modifying values of this one pair triggers modification of the interpolated values for pairs with neighbor base parameter values. By moving the object to another position, both values of the value pair, the base parameter value (e.g., horizontal position) and/or the state value (e.g., vertical position), can be modified by the user, thus resulting in a modified configuration. It is to be noted that the user can also provide input by modifying multiple values of multiple configuration value pairs (e.g., by performing a respective sequence of drag actions).

[0014] The default configuration is provided as an input to the optimizer component. The optimizer component includes at least one optimization algorithm for the selected base parameter/state type under the given constraint. As different base parameter/state type pairs can be supported the optimizer component may include a plurality of optimization algorithms dedicated to various combinations of base parameters/state types. As explained above, a particular configuration (i.e., a particular plurality of configuration value pairs) can be achieved with multiple (different) combinations of real world inputs. Such input

combinations are referred to as input combinations “consistent with a given configuration”. As further explained above, the set of such input combinations is given by a semi-analytic set depending on the chosen configuration. For a given configuration, the optimizer component is optimizing an objective across all possible input combinations that are consistent with the given configuration. The objective depends on the exact properties of the physical entity in question. For many physical entities, a change in a single input may impact the behavior of the physical entity in multiple future states because of the linkages between potential states; the optimizer component accounts for these linkages by using the explicit characterization of the semi-analytic set of input combinations consistent with the given configuration. This explicit characterization must be derived in terms of explicit equalities and inequalities involving explicit real analytic functions. For example, a user’s objective could be to minimize non-renewable energy consumption for a given configuration. That is, given the chosen configuration, the objective is to optimize over the semi-analytic set of input combinations that are consistent with the given configuration and satisfy the admissibility constraints in order to find the input combination that attains minimal consumption of non-renewable energy. The underlying optimization algorithm uses the explicit characterization of the semi-analytic set of input combinations through explicit equalities and inequalities with real analytic functions. Potentially, even for the same base parameter/state type combination different optimization algorithms may be available to address different optimization constraints. The optimizer component now applies the respective optimization algorithm (i.e. the optimization algorithm adapted to optimize the default configuration under a predefined constraint) to the default configuration and provides at least one graphical result object to the user. The graphical result object represents at least one corresponding result value characterizing the outcome of the objective optimization (i.e., the optimization of the objective) for the default configuration over the semi-analytic set of input combinations consistent with the default configuration and satisfy the admissibility constraints. In the example of the graphical representation of the potential states as a function of the base parameter the single value may be represented as a horizontal line reflecting the value as a constant in the graph or as any other representation suitable to represent a numeric value. Typically, the result value is a single value. However, in some situations multiple result values may be computed by the optimization algorithm. For example, when consumption of two

different types non-renewable energy (e.g., oil and wood) are to be minimized, the graph may show the consumption of both energy types.

[0015] Again, without any intention to limit the scope of the claims, the electrical device example is used for explanation. The electrical device consumes renewable energy when available, and non-renewable energy otherwise. Given a configuration of indoor and outdoor temperatures, there may be many ways to achieve this configuration using different types of energy for temperature control. The optimization problem over all input combinations consistent with a given configuration and satisfying the admissibility constraints faces a tradeoff between the use of renewable (e.g., solar) energy for operation or storing this energy in a battery for future needs (e.g., for situations in which the lighting conditions (with daylight) get really bad). This is a non-trivial optimization problem. The problem becomes even more complicated when, in addition, wind energy is taken into account. Any choice of an input impacts the behavior of the physical entity in multiple future states, and the optimizer takes these linkages into account. For example, changing a temperature of the smart house in one state naturally impacts temperature in other states because temperature adjusts slowly. Mathematically, the linkages across states are accounted for by the explicit equalities and inequalities that define the set of input combinations consistent with a given configuration. Thus, by accounting for these equalities and inequalities, the algorithm also accounts for linkages across states.

[0016] For the predefined configuration, the optimizer component solves the optimization problem above with a respective optimization algorithm across all possible inputs that are consistent with the predefined configuration and visually represents the expected attained objective as a single value. That is, the user can immediately see how the current configuration affects the future state of the physical entity, accounting for the linkages across potential states and the complex nature of equalities and inequalities that define the set of input combinations that are consistent with the given configuration. In other words, the graphical result object reflecting the result value provided by the optimizer presents to the user a characteristic value that provides aggregated information about all potential future states of the physical entity and about the best possible way of achieving this configuration. For example, if the objective is to minimize non-renewable energy consumption, averaged across all possible future states of the physical entity, the graphical

result object precisely reports this average value, thus summarizing numerous explicit (i.e., given by the properties of the optimal input combination selected by the optimization algorithm) and implicit (i.e., those input combinations that were rejected by the algorithm because of their sub-optimality) characteristics of the configuration into one number. In an optional embodiment, the graphical user interface may provide a drill-down function which allows the user to select the graphical result object to retrieve further information about the optimization. For example, such further information may include the optimization inputs which were used to achieve the current (or default) configuration. Such optimization inputs are otherwise hidden from the user. For example, an optimization input may be the energy consumption of a device. Further examples are illustrated in the detailed description. The optimizer component receives such inputs either as additional optimization inputs from a user or the inputs may be predefined in an appropriate data structure from where they can be retrieved automatically. Examples of such additional inputs can be: The smart house is powered by solar energy, wind energy and fossil energy sources (e.g., oil or gas), or: A data structure processing algorithm can be assigned certain number of CPU/GPU and memory resources. An electrical device may be designed using a certain number of relays with different activation thresholds.

[0017] The user pursues the technical task to define a configuration for the physical entity which results in a further optimized configuration of states across future values of the base parameter when compared to the current (or default) configuration. To achieve this goal the user interacts with the graphical user interface to modify the current configuration by applying a drag action for at least one of the interactive graphical user interface objects. Such a drag action is sensed by the user interface device and changes the potential state associated with the respective interactive user interface object (e.g., changes the state value of the respective interactive user interface object for one or multiple potential future values of the base parameter). For example, the state value may be moved to a higher or to a lower value by touching on the interactive object and dragging it upwards or downwards on the screen. In another implementation, a slider may be used to apply the movement to the interactive object once the object is selected (e.g., by a mouse click or the like). Further, the corresponding base parameter value may be moved to a higher or to a lower value by selecting the interactive object and dragging it left or right on the screen. Of course, both values of the configuration value pair can be changed simultaneously when dragging the

interactive user interface object in a diagonal direction or any direction which deviates from a mere horizontal or vertical direction. When the user drags the interactive object, the modified state/base parameter value which corresponds to the new position of the interactive object is determined by the graphical user interface via the respective display coordinates. In other words, by executing the drag action the user provides a user defined input to modify at least one associated default configuration value pair into a user defined modified configuration value pair. This user action results in at least one modified potential state of a user defined configuration.

[0018] The modified configuration value pair(s) is/are received by the optimizer component and the optimization algorithm is re-applied to the modified configuration. The new, modified configuration defines a new semi-analytic set of input combinations that are consistent with the new configuration. Given the explicit characterization of this semi-analytic set by explicit equalities and inequalities, the optimization algorithm runs over this semi-analytic set to find the optimal combination of inputs consistent with the modified configuration under the respective admissibility constraints. Again, the optimization algorithm determines at least one result value characterizing the outcome of the optimization for the user defined modified configuration over the new semi-analytic set. At least one corresponding modified graphical result object (representing this result value) is then presented to the user. By comparing the value corresponding to the initial graphical result object with the value corresponding to the modified result object, the user can immediately recognize whether the modified configuration leads to an improved configuration of a future state of the physical entity across a set of possible future values of the base parameter (i.e., across the base parameter values of all potential states or a subset of those) compared to the previous configuration. Even if the user modifies the configuration for just one configuration value pair, such an action may impact the behavior of the physical entity in multiple states due to linkages between these states. A human user is typically unaware of all the linkages between potential states. The graphical result objects effectively summarize the impact of said linkages on the objective into corresponding values, thus reducing the degree of complexity to a level which can be handled by a human user.

[0019] The modification of the configuration can be repeated multiple times until the user is satisfied with the degree of optimization achieved by a particular modified configuration.

The user finally selects the particular modified configuration (or the initial default configuration) to operate the physical entity in accordance with the selected configuration. The optimizer then selects the optimal combination of inputs among all possible input combinations consistent with the selected configuration and satisfying the admissibility constraints by running the optimization over the semi-analytic set defined by the selected configuration and using the explicit characterization of this set in terms of equalities and inequalities. This combination of inputs is then implemented for the physical entity.

[0020] Referring again to the electrical device example, the user defined modified configurations with adjusted potential state values may take into account the user's beliefs about future weather conditions (light intensity) and the importance of renewable energy for the user. The importance of the renewable energy is reflected in the user defined optimization constraint and becomes part of the objective function for the optimization algorithm.

[0021] In one embodiment, a checker component of the computer system may be used to evaluate if a modified configuration is executable under real-world conditions before the modified configuration is deployed to the physical entity. Configurations failing this consistency check may be rejected by the GUI right away as non-deployable configurations. As used herein, the configuration is considered to be executable under real-world conditions if it can be achieved under normal circumstances with existing technologies and accounting for linkages between states. A configuration is executable under real-world conditions if and only if there exists a non-empty set of input combinations consistent with this configuration and satisfying the inequality constraints. Thus, the consistency checker verifies that the corresponding semi-analytic set is non-empty. For example, if the base parameter is outdoor temperature and the state is indoor temperature, then a configuration then a configuration (20, 20) and (20.1, 40) is not executable under normal circumstances because the two states are linked: the outdoor temperature can move from 20 to 20.1 degrees Celsius in less than a minute, while it is not possible to heat the house from 20 to 40 degrees over such a short period of time. Similarly, a state (20, -20) is not executable because one cannot achieve an indoor temperature of -20 if the outdoor temperature is 20. The set of executable configurations can always be defined in terms of an explicit set of equalities and inequalities and can thus itself be represented as a semi-analytic variety.

[0022] Further aspects of the invention will be realized and attained by means of the elements and combinations particularly depicted in the appended claims. It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only, and are not restrictive of the invention as described.

Brief Description of the Drawings

[0023]

FIG. 1 includes a simplified block diagram of a computer system for editing a configuration of a physical entity according to an embodiment;

FIG. 2 is a simplified flow chart illustrating a computer-implemented method editing a configuration of a physical entity to optimize a future state of the physical entity according to an embodiment;

FIGs. 3 to 4 show simplified views of graphical user interfaces according to various embodiments;

FIG. 5 shows a further simplified view of a graphical user interface according to an embodiment; and

FIG. 6 is a diagram that shows an example of a generic computer device and a generic mobile computer device, which may be used with the techniques described herein.

Detailed Description

[0024] FIG. 1 includes a simplified block diagram of a computer system 100 with a graphical user interface GUI 200 for editing a configuration of a physical entity 30 according to an embodiment. FIG. 1 will be described in the context of the flow chart of FIG. 2 illustrating a respective computer method 1000 for editing the configuration according to an embodiment. For this reason, the following description includes reference numbers for both figures. The following description is explaining the embodiments of the invention by way of example. The chosen example of a building temperature regulation as illustrated in FIG. 3 is not intended to limit the scope of the claims to physical entities which are buildings (e.g., smart houses). Rather, additional examples will show that the inventive concept is applicable

to any physical entity operable according to a configuration describing its potential future states.

[0025] The computer system 100 for editing a configuration of the physical entity 30 supports the user 10 to optimize a future state of the physical entity 30. The configuration includes (or corresponds to) a plurality of potential states for the physical entity 30. Each potential state corresponds to a configuration value pair which includes a particular value of a base parameter and a particular value of an associated state value. In other words, the state values SV1 to SV4 of the four potential states in the example of FIG. 1 depend on the respective base parameter values PV1 to PV4 and the state values can be described as a function f_1 of a base parameter BP.

[0026] The computer system 100 has a graphical user interface component GUI 200. GUI 200 provides standard input/output functions (e.g., to support interaction via touch screen, mouse, keyboard or the like) to the user to let the user interact with the computer system. In an optional initialization step the GUI 200 receives a selection for the base parameter BP and the type PST of the potential states from the user 10. With this selection, the user can set the context for the computer system. Alternatively, the system may be pre-configured with a particular configuration scenario where the base parameter BP and state type PST is already known by the system (e.g., predefined BP and PST).

[0027] GUI 200 provides to the user 10 an initial visual representation of a default configuration which corresponds to a plurality of configuration value pairs for the potential states forming part of the default configuration. Each of the potential states is represented by a corresponding interactive graphical user interface object PS1 to PS4 which represents the respective default function value of the potential state at the associated base parameter value.

[0028] Turning now briefly to FIG. 3 to illustrate a graphical user interface 300 used for temperature regulation in a smart house. The physical entity in this example can be considered to be the heating of the smart house. In the example, the interactive user interface objects are represented by diamond icons. In general, an interactive user interface object may have any appropriate shape (e.g., square, rectangle, circle, etc.) which allows the user to distinguish the interactive object from other graphical user interface objects. The

interactive objects may also be distinguished from other objects by other display properties, such as for example, the color, background pattern or even the display mode (e.g., highlighting by blinking, etc.) The black diamonds DS1 to DS4 in the example belong to the default configuration. In this scenario, the base parameter BP is the outside temperature OuT expected during the winter months. The type PST of the state values is selected as the temperature InT inside the smart house. In the example, possible values of the base parameter (outside temperature, OuT) are $OuT = -5, 0, 5, 10$ degrees Celsius. Of course, in a real application many more temperature values might be considered to have higher granularity for the temperature scale (e.g., a base parameter value for a potential state may be defined at every half degree) and to have a temperature range with lower min and higher max temperatures.

[0029] The configuration describes the average internal (room) temperature as a function of the temperature outside. The default configuration suggests a constant inside temperature InT of 20 degrees Celsius for all potential states. In other words, the heating of the smart house would be controlled by the default configuration to keep the room temperature at 20 degrees independent of the outside temperature.

[0030] Turning back to FIG. 1, the optimizer 120 receives the default configuration (e.g., via the interface 110 and performs an optimization over all combinations of optimization inputs that are consistent with the default configuration and satisfy the admissibility (inequality) constraints. Thereby, the optimizer can take into account one or more optimization constraints on the set of admissible inputs (i.e., the set of input combinations that are implementable in the real world under normal physical conditions; or constrains imposed by the user due to some user-specific needs and requirements. For example, the indoor temperature cannot be above 100 degrees Celsius under normal conditions). An optimization constraint is an additional input into the optimization algorithm. The optimization constraint may be modifiable (e.g., a user can set an upper bound on the admissible energy consumption). The default configuration may be initially set by the user or it may be a preconfigured set of potential future states (e.g., stored in a data structure used for operating the physical entity).

[0031] In one embodiment, the consistency checker component 130 is integrated into the graphical user interface, and does not permit the user to select configurations that are not implementable in the real world. The consistency check takes into account constraints imposed by the real world and can be performed by a checker component to evaluate if a modified configuration overall is executable under real-world conditions: Usually, such evaluation reduces to verifying that some observable physical characteristics of the physical entity and its environment satisfy certain equality and inequality constraints. Configurations failing this consistency check may be rejected by the GUI right away as non-deployable configurations. In such cases, the GUI may automatically switch back to the optimal configuration (e.g., by performing an undo action on the previous drag action) and inform the user, that the intended modification of the configuration is not allowed. That is, the system may further include a checking function that verifies whether a user modified configuration is feasible, and sends an error message if the configuration is not feasible under real-world conditions.

[0032] In the smart house heating scenario of FIG. 3, examples for constraints are: the house can be heated and cooled using water (which in turn is heated using gas, oil, or wood, referred to as GOW), or by electric air conditioning powered by regular electricity referred to as RE. The term regular electricity as used herein relates to electricity which is generated from any arbitrary mix of energy sources including nuclear energy, or energy produced by coal or gas driven power plants. The objective of the optimization is to reduce the consumption of non-renewable energy used for the heating of the smart house to reduce damage to the environment. The damage to the environment is considered to increase with the total use of energy by the smart house heating. In other words, in the example, energy is differentiated into renewable and non-renewable energy and the optimization goal is to prefer to use of renewable over non-renewable energy.

[0033] Suppose that the outdoor temperature, $OuT(t)$ follows a Markovian stochastic process as a function of time t . Suppose also that the changes in the interior temperature, $InT(t)$, are determined by the difference between the indoor and the outdoor temperature, $InT(t)-OuT(t)$, as well as the two types of energy consumption, $GOW(t)$ and $RE(t)$.

$$d(InT(t)) = G1(InT(t)-OuT(t), GOW(t), RE(t)) dt \quad (1)$$

Here, G_1 is a function that is decreasing in its first argument and decreasing in the other two arguments. The differential equation (1) characterizes the linkages between the states of the physical entity in the example. Namely, states with different values of OuT are linked because temperature adjusts slowly. Thus, an action to impact temperature for one value of the base parameter influences the state of the physical entity for other values of base parameters, thus impacting the whole configuration. These linkages account for a high degree in complexity of the optimization.

[0034] A state configuration $CONF = ((InT_1, OuT_1), (InT_2, OuT_2), (InT_3, OuT_3), (InT_4, OuT_4))$ is represented by the piece-wise linear curve $InT = PSL(OuT)$ where PSL is the corresponding piece-wise linear function in GUI 300 with the interactive objects DS_1 to DS_4 . The input combination $INPUT_COMBIN$ is given by functions $GOW = W_1(InT, OuT)$ and $RE = W_2(InT, OuT)$. This input combination is consistent with $CONF$ if - and only if the following consistency conditions are fulfilled:

$$PLS(OuT(t)) - bound < InT(t) < PLS(OuT(t)) + bound \quad (\text{consistency conditions})$$

holds for all values of time t .

Here, *bound* is a user-specified tolerance for the deviation of the temperature InT from the target configuration temperature $PLS(OuT)$.

Suppose also that the damage to the environment associated with $INPUT_COMBIN$ is given by a function F_1 :

$$F_1(INPUT_COMBIN) = F_1(GOW, RE) = F_1(W_1(InT, OuT), W_2(InT, OuT))$$

The nature of this function can be complex.

The objective can thus be:

Objective = minimize expected environmental damage

$$E[\int_0^T F_1(W_1(InT(t), OuT(t)), W_2(InT(t), OuT(t))) dt]$$

over all input combinations satisfying the consistency conditions.

[0035] The optimization algorithm OA1 (cf. FIG. 1) solves this minimization problem. The graphical result object DRO can represent the result of this minimization. For example, DRO may include a numerical value illustrating the energy consumption associated with the default configuration. For example, the graphical result object DRO may represent the average energy consumption $E[\int_0^T (W1(\text{InT}(t), \text{OuT}(t)) + W2(\text{InT}(t), \text{OuT}(t)))dt]$ for the energy profile that solves the minimization problem of the “Objective” function. The mathematical expectation, $E[]$ can be computed using techniques from probability theory such as partial differential equations, maximum likelihood estimation, Bayesian learning, linear- and non-linear filtering, etc.

[0036] In the example, the eight numbers $((20, -5), (20, 0), (20, 5), (20, 10))$ represent the default configuration of (InT, OuT) . The corresponding optimal input combination consistent with the default configuration is given by the two functions $W1(\text{InT}, \text{OuT})$, $W2(\text{InT}, \text{OuT})$ and are typically not shown to the user via GUI 200. Given the default configuration of (InT, OuT) , the user wants to see a graphical result object representing the result of the “Objective” optimization. It is to be noted, that in this example the outcome of the optimization $E[\int_0^T F1(W1(\text{InT}(t), \text{OuT}(t)), W2(\text{InT}(t), \text{OuT}(t)))dt]$ is not necessarily shown to the user. Instead, other characteristics of the configuration could be shown in the GUI 200, given by some (linear or non-linear) function of the optimal input combination consistent with the given configuration.

[0037] Persons skilled in the art of optimization are able to select an appropriate optimization algorithm (e.g., standard gradient descent methods, dynamic programming (Bellman and Hamilton-Jacobi-Bellman equations) or combinatorial optimization methods) for solving the optimization problem associated with the respective example scenario.

[0038] In general, a result value is determined by the optimization algorithm OA1 which is the outcome of solving the “Objective” minimization problem over all possible inputs (INPUT_COMBIN) consistent with the given configuration and satisfying the admissibility constraints. The GUI 200 provides 1200 to the user 10 a graphical result object RO1 representing this result value. In the example of FIG. 1, the result object RO1 is a line visualizing the result value whereas in the example of FIG. 3 the result object includes a numerical representation DRO of the result value. The result object representation as

shown in FIG. 1 may be preferable in cases where the result value can be expressed in the same units of measurement as the potential state values. In other words, if the outcome of the optimization has the same dimension as the type of the potential states in the interactive GUI, the value associated with the graphical result object can be fit into the respective GUI presentation as a constant value represented by a horizontal line.

[0039] Once the user is confronted with the optimization result for the default configuration, the user may be concerned about the energy consumption associated with the default configuration, and may start modifying the default (or current) configuration. The modification occurs via a drag action. In FIG. 1, the interactive object PS4 (SV4, PV4) is moved by the drag action DA1 to define a new potential state represented by the interactive user interface object PS4'. PS4' is associated with the configuration value pair (SV4', PV4'). This drag action DA1 has also modified the function f1 of the default configuration into a function f1' characterizing the modified configuration.

[0040] In the example of FIG. 3, the user drags the internal temperature values associated with the low outside temperatures -5 and 0 degrees to 18 degrees and thus generates modified potential states represented by the diamonds DS1', DS2'. This data input action results in a modified configuration corresponding to the four configuration value pairs (18, -5), (18, 0), (20, 5), and (20, 10) which is now received 1300 by the optimizer.

[0041] The incentives to modify the default configuration may be user-specific and may be difficult to model algorithmically. Hence, it is crucial for the user to have the ability to personally interact with the GUI and see the impact of user choices on the various characteristics the user cares about. For example, when the user is concerned about the energy consumption, the user may be ready to accept a lower room temperature when it is very cold outside. This is reflected by the modified configuration. The optimization algorithm of the optimizer is now re-applied 1400 to the modified configuration further optimizing the combination of inputs consistent with the future physical entity (e.g., house heating) state in accordance with the modified configuration under the admissibility constraints, over the semi-analytic set of inputs that are consistent with the given configuration. The optimizer again determines a result value for the modified configuration which is then provided to the GUI 200 to be visualized, and is also provided 1500 to the user 10 as graphical result object

RO'. The user immediately recognizes by how much the optimal energy consumption of the smart house is reduced when applying the modified configuration with the potential states represented by DS1', DS2', DS3 and DS4. Then, the user is able to gauge the optimal energy saving relative to his personal feelings about the loss of comfort due to a lower room temperature. Due to the linkages between states, changing temperature in one state may have an impact on the energy consumption for all other states. The immediate feedback that the user gets from observing the change in the graphical object caused by the change in configuration serves as an efficient, aggregated graphical representation of these complex linkages which otherwise were hidden from the user and are mathematically expressed in terms of the equalities and inequalities defining the semi-analytic set of input combinations that are consistent with a given configuration. It therefore supports the user to take an informed decision taking into account the impact of the modified configuration with regards to the objective of the optimization.

[0042] In case the user is not impressed by the energy savings achieved by the modified configuration, but is rather afraid of being cold when it is very cold outside ($OuT=-5$), the user may perform a further drag action to move DS2' back to the position of the original DS2 object, and then further move the DS1' object to a new position (DS1'') with the configuration value pair (22, -5). Again, this further modified configuration will become an input to the optimizer 120 and the optimization algorithms (e.g., OA1 and/or OA2) determine one or more further result values being provided 1500 to the user, for example, as the graphical result object RO''.

[0043] If the user is satisfied by the tradeoff between the modified energy consumption and the modified assessment of comfort from the new configuration (for example, when the additional expected feeling of comfort from higher room temperatures over-weights the disutility from higher energy consumption), the user can select 1600 this modified configuration and have it deployed 1700 to the house heating control unit in order to operate the physical entity (house heating) in accordance with the selected configuration.

[0044] Thus, an advantageous feature of the GUI is its ability to provide a simple and intuitive visualization of how this tradeoff affects the future states of the physical entity, combined with a (sophisticated) hidden layer of optimization that accounts for linkages

across states. In other words, the graphical result objects reflect the (expected) internal state of the physical entity associated with the respective configuration. Deployment of the configuration in the example of FIG. 3 means that the configuration value pairs are transmitted to a controller of the house heating device where they are used for temperature regulation in accordance with the configuration value pairs of the selected configuration.

[0045] In general, as shown in FIG. 1, the user can drag DA1 an interactive user interface object PS4 in such a way that both values of the respective configuration value pair are modified. In the example of FIG. 1, the potential state (SV4, PV4) is the one which is important to the user and the result of the optimization applied to the modified configuration is represented by the second graphical result object RO2 giving an immediate feedback to the user regarding the implication of the user driven modification of the configuration on the physical entity 30. In particular, this immediate feedback serves as an efficient tool for visualizing the (potentially complex) linkages between states. In the example, the configuration c1 is finally selected by the user and deployed to the physical entity.

[0046] Optionally, the optimizer 120 may include a plurality of optimization algorithms. For example, the further optimization algorithm OA2 may be suitable for a completely different optimization scenario with a different base parameter BP and a different type PST of potential states as described in the example of FIG. 4.

[0047] It is to be noted that, in one embodiment, the user may also delete any one of the existing potential states in a configuration, or may add additional potential states to the configuration. This embodiment will be described in more detail in FIG. 6.

[0048] FIG. 4 illustrates a further simplified view 400 of a GUI according to an embodiment. The example scenario behind FIG. 4 is similar to the example scenario explained with FIG. 3. An owner of an in-door swimming pool decides about the water temperature. However, in addition to the existing GOW heating options the heating energy can also be solar energy SE. For this example scenario, the base parameter is solar irradiance SI (that is, solar light intensity). The decision to be taken by the pool owner is about the temperature configuration depending on the value of the SI. Such a configuration can be implemented in

different ways by storing produced solar energy in a battery (or another appropriate energy storage), or by using it for heating.

[0049] Therefore, the potential states in this example are characterized by the water temperature (WT). The heating system has 4 Type-1 heater devices, indexed by $J = 1, 2, 3, 4$. Heater device number J is characterized by a threshold $K(J)$ and a power $P(J)$ both of which are adjustable. Given a pair of parameters (K, P) , the corresponding heating device gets activated when $SI < K$ and starts heating up, producing a temperature increase of $(K - SI)^+ * P$, where we have defined $(K - SI)^+ = \max(K - SI, 0)$. In addition, other heating devices are available (henceforth, Type-2), with an upper bound on their heat production, given by $\min(k, (K - SI)^+) * Q$. That is, heat production of such a device is capped at $k * Q$. For simplicity, we assume that there are three such devices, $\min(K(1), (K(2) - SI)^+) * Q(1)$, $\min(K(2), (K(3) - SI)^+) * Q(2)$, $\min(K(3), (K(4) - SI)^+) * Q(3)$. We also assume that, absent heating, WT is linear in SI: $WT = a + b * SI$.

[0050] Thus,

INPUT_COMBIN = (($K(1)$, $P(1)$), ($K(2)$, $P(2)$), ($K(3)$, $P(3)$), ($K(4)$, $P(4)$), $Q(1)$, $Q(2)$, $Q(3)$)

Given the INPUT_COMBIN, the temperature configuration (water temperature as a function of solar irradiance) is given by

$$WT(SI) = a + b * SI + \sum_{j=1}^4 (K(j) - SI)^+ * P(j) + \sum_{j=1}^3 \min(K(j), (K(j+1) - SI)^+) * Q(j)$$

Clearly, this is a piece-wise linear function with potential kinks at $SI = K(1), \dots, K(4)$.

[0051] The non-linear nature of heating devices introduces complex linkages between states: a choice of a trigger $K(j)$ and intensity $P(j)$ influences water temperature in many future possible states. Thus, grasping the impact of the choice of one single parameter on the whole configuration is not possible for a human user.

[0052] Similar to the previous example, the user intends to deploy an optimized configuration for temperature regulation of the pool water temperature. It is assumed that the default configuration includes again a constant water temperature configuration for all four potential states at each of the following four base parameter values: $SI1 = 50 \text{ W/m}^2$, $SI2 = 100 \text{ W/m}^2$, $SI3 = 200 \text{ W/m}^2$, $SI4 = 300 \text{ W/m}^2$. In this example, the base parameter values

coincide with the trigger values $K(1), \dots, K(4)$ of the heating device. The corresponding potential states are represented by the piece-wise linear function connecting the interactive diamond objects ES1 to ES4, respectively.

[0053] Thus, the current configuration is presented to the user in the interactive GUI 400 for selection or modification:

$$\text{CONFIG} = ((\text{WT}(1), \text{SI}(1)), (\text{WT}(2), \text{SI}(2)), (\text{WT}(3), \text{SI}(3)), (\text{WT}(4), \text{SI}(4)))$$

Thus, the GUI 400 initially shows the interactive objects ES1 to ES4 with the default configuration value pairs (22, 50), (22, 100), (22, 200), (22, 300).

An input combination $\text{INPUT_COMBIN} = ((K(1), P(1)), (K(2), P(2)), (K(3), P(3)), (K(4), P(4)), Q(1), Q(2), Q(3), Q(4))$ is consistent with a given configuration

$$\text{CONFIG} = ((\text{WT}(1), \text{SI}(1)), (\text{WT}(2), \text{SI}(2)), (\text{WT}(3), \text{SI}(3)), (\text{WT}(4), \text{SI}(4)))$$

if and only if $\text{SI}(1)=K(1), \dots, \text{SI}(4)=K(4)$, and

$$\text{WT}(i) = a + b * \text{SI} + \sum_{j=1}^4 (K(j) - \text{SI}(i))^+ * P(j) + \sum_{j=1}^3 \min(K(j), (K(j+1) - \text{SI}(i))^+ * Q(j) \quad (\text{consistency conditions})$$

for all $i=1,2,3,4$.

In addition, both Q and P must be nonnegative, and satisfy upper bounds $Q < Q_{\max}$, $P < P_{\max}$, where $P_{\max}$ and $Q_{\max}$ are constants defined by the user and might originate from regulatory or legal requirements. The consistency conditions define an explicit convex polytope in the 7-dimensional space of (Q, P) parameters. This convex polytope is the semi-analytic set defined by the configuration. Its defining explicit equalities and inequalities serve as inputs to the optimization algorithm.

We assume that each device of Type-1 with parameters (P, K) consumes $F(P, K)$ non-renewable energy on average, for some function $F(P, K)$, while each device of Type-2 with parameters (Q, K) consumes $G(Q, K)$ non-renewable energy on average for some function $G(P, K)$. The user objective is to minimize total non-renewable energy consumption

$$\sum_{j=1}^4 F(P(j), K(j)) + \sum_{j=1}^3 G(Q(j), K(j))$$

over all input combinations satisfying the consistency conditions. For example, standard gradient descent methods or combinatorial optimization methods may be used to find the optimal INPUT_COMBIN. A person skilled in the art may also use other appropriate optimization methods. In this example, the semi-analytic set defined by the consistency conditions and the admissibility constraints reduces to an explicit convex polytope.

[0054] Again, the dependence of the objective function on the INPUT_COMBIN can be complex because of the highly complex linkages between states produced by the non-linear nature of the heating devices. Mathematically, it is represented by the complex structure of the above convex polytope. This polytope depends on the chosen configuration in a very complex way. Every modification of the configuration is reflected in a modification of the convex polytope and then also in the outcome of the optimization over this polytope.

The system can provide the outcome of this minimization problem as a single graphical result object ERO representing the total energy consumption. However, the user may also want to see a breakdown into $\sum_{j=1}^4 F(P(j), K(j))$ and $\sum_{j=1}^3 G(Q(j), K(j))$. In this case, the GUI 400 can also provide to the user two corresponding additional graphical result objects associated with the default configuration.

[0055] The user may now modify the default configuration by deciding that in situations with more available solar energy (higher SI values) a warmer pool water temperature is intended. In the example, the user drags the interactive objects ES3 and ES4 to new positions ES3' and ES4'. As a result, the modified configuration provides the modified configuration value pairs (24, 200) and (25, 300) as input to the optimizer. That is, the user intends to see the impact on the energy consumption values when a higher pool temperature (24 and 25 degrees) is set in case of more available solar energy (200 W/m² and 300 W/m²). The optimization algorithm is now re-applied to the modified configuration. The new convex polytope is computed for the modified configuration, the optimization is run over this polytope, and the respective graphical result object(s) ERO' (and optionally GOW', FSE') are shown to the user. The user can now select the default or the modified configuration for deployment. In case the additional energy consumption for the modified configuration is primarily compensated by the solar energy, the user may decide to deploy the modified configuration to have a more convenient water temperature while still keeping

low the consumption of non-renewable energy. Again, the decision about the choice of the optimal configuration is fully determined by the user-specific assessment of the tradeoff between additional comfort and the unhappiness about additional energy consumption. The GUI provides a simple and intuitive visualization of this tradeoff, combined with a (sophisticated) hidden layer of optimization which allows the user to identify the desired configuration to be deployed to the physical entity without worrying about the mathematical nature of the convex polytope and the optimal choice of the input combination. Thereby, the user is supported by the claimed interactive data input method allowing to modify the configurations and finally selecting a desired modification. In addition, the CO₂ emission value associated with the energy consumption may be computed. However, for the disclosed interactive user input method for editing configurations this is of no relevance at all. In general, the values represented by the graphical result objects can be determined by an arbitrary function of the input combinations. That is, the computer system provides future status information regarding a technical system (e.g., pool with heating), and the information provides guidance for that the system can be optimized (e.g., using or not using energy from a particular source, etc.)

[0056] FIG. 5 is based on the example of FIG. 3 illustrating how a user can use the GUI 300 to also modify the granularity of a configuration by adding or deleting potential states which are then used as input for the respective optimization. In the example, the default configuration corresponds to the default configuration of FIG. 3. The user now decides to have a higher granularity for the heating control by modifying the default configuration in the following way. As in FIG. 3, the user increases the state value associated with DS1 to result in DS1''. Further, the state DS2 is modified into DS'''. In addition, the user is inserting additional potential states DSi1'', DSi2'', DSi3'' and deletes the original state DS3 (the deletion marked as DSd3). For the insertion and deletion of interactive graphical objects, any known methodology can be used which allows the user to add or delete graphical objects in GUI 300. The modified configuration then includes the interactive objects DS1'', DSi1'', DS'', DSi2'', DSi3'', DS4, and serves as a new input for the optimization algorithm(s).

[0057] For example, the user may touch on a certain point of the GUI 300 and indicate that a new potential state is to be inserted at the respective position. For example, a context menu may provide the option to insert such an object. This insert action performed by the user can

be sensed by the user interface device and, thereby, a corresponding user defined input for inserting a further graphical interactive user interface object as a further potential state of the user defined modified configuration is received. The configuration value pair associated with the inserted object (i.e., the associated base parameter and state value) can then be derived by the system from the position of the inserted object in the graphical user interface.

[0058] The user may also remove existing potential states from a configuration. For example, the user can select a respective interactive object (e.g., by touching on it) and select a delete action (e.g., from a context menu). As a result, the system removes the selected object from the corresponding configuration.

[0059] It is to be noted that the previous examples only have explanatory character and are not supposed to limit the scope of the claims with regards to the types of physical entities described in the example. Rather, the claimed data input approach for modifying configurations of physical entities can be applied to any physical entity which may be subject to configuration.

[0060] Another example of a physical entity could be a control unit of a computer system executing a computer algorithm that activates additional CPU to speed up computations in the computer system. In this example, the base parameter is the number of computation tasks (NT) submitted to the computer system, and the state type is the average speed (AS) with which these tasks are processed. According to the example, there are several processing units (CPU) that get activated when NT exceeds a threshold k , and reach their maximal capacity K when NT exceeds $K+k$.

[0061] That is, for each such unit, $\text{CPU}(NT) = \min(K, \max(NT-k, 0))$. Q is used to denote the number of such units deployed. A configuration in this example is given by four configuration value pairs $(AS(1), NT(1)), \dots, (AS(4), NT(4))$. Then the average speed $AS(NT)$ is given by a piece-wise linear function interpolating the four configuration value pairs. Given an input combination $((k(1), K(1), Q(1)), \dots, ((k(n), K(n), Q(n))))$, this combination is consistent with the configuration if, and only if, all $k(i), K(i), i=1, \dots, n$ belong to the set $(NT(1), \dots, NT(4))$ and

$$AS(j) = \sum_{i=1}^n Q(i) * \min(K(i), \max(NT(j) - k(i), 0)), j=1,2,3,4 \quad (\text{consistency conditions})$$

In addition, the Q parameters may satisfy the admissibility constraints, $Q(i) > 0$ and $Q(i) < Q_{\max}$, where $Q_{\max}$ is a constraint imposed by the user that may originate in requirements from a particular domain.

The non-linear nature of these consistency conditions implies that there are linkages across states: a modification of a single input changes the whole configuration (i.e., all potential states of the configuration). However, mathematically, the set of Q parameters is a convex polytope in the n-dimensional Euclidean space, defined explicitly by the equality constraints (consistency conditions) and the inequality constraints (admissibility conditions).

It is assumed that average CPU usage of a CPU of a processing unit with characteristics (k, K) is given by $G(k, K)$, and the objective is to minimize the usage

$$\text{Minimize } \sum_{i=1}^n Q(i) * G(k(i), K(i))$$

over all INPUT_COMBIN satisfying the consistency conditions. These consistency conditions reduce to a simple linear system for the coefficients Q, and hence a minimization problem under a system of linear equality and inequality constraints is solved. Thus, with respect to Q-inputs, optimization can be solved using standard linear programming methods (e.g., the simplex method).

[0062] FIG. 6 is a diagram that shows an example of a generic computer device 900 and a generic mobile computer device 950, which may be used with the techniques described here. Computing device 900 is intended to represent various forms of digital computers, such as laptops, desktops, workstations, personal digital assistants, servers, blade servers, mainframes, and other appropriate computers. Generic computer device may 900 correspond to the computer system 100 of FIG. 1. Computing device 950 is intended to represent various forms of mobile devices, such as personal digital assistants, cellular telephones, smart phones, and other similar computing devices. For example, computing device 950 may be a smartphone or tablet computer which serves as the GUI 200 of FIG. 1. In some embodiments, the computing device 950 may also include the optimizer. In other embodiments, the optimizer may be implemented by the generic computer device 900. The components shown here, their connections and relationships, and their functions, are meant

to be exemplary only, and are not meant to limit implementations of the inventions described and/or claimed in this document.

[0063] Computing device 900 includes a processor 902, memory 904, a storage device 906, a high-speed interface 908 connecting to memory 904 and high-speed expansion ports 910, and a low speed interface 912 connecting to low speed bus 914 and storage device 906. Each of the components 902, 904, 906, 908, 910, and 912, are interconnected using various busses, and may be mounted on a common motherboard or in other manners as appropriate. The processor 902 can process instructions for execution within the computing device 900, including instructions stored in the memory 904 or on the storage device 906 to display graphical information for a GUI on an external input/output device, such as display 916 coupled to high speed interface 908. In other implementations, multiple processing units and/or multiple buses may be used, as appropriate, along with multiple memories and types of memory. Also, multiple computing devices 900 may be connected, with each device providing portions of the necessary operations (e.g., as a server bank, a group of blade servers, or a processing device).

[0064] The memory 904 stores information within the computing device 900. In one implementation, the memory 904 is a volatile memory unit or units. In another implementation, the memory 904 is a non-volatile memory unit or units. The memory 904 may also be another form of computer-readable medium, such as a magnetic or optical disk.

[0065] The storage device 906 is capable of providing mass storage for the computing device 900. In one implementation, the storage device 906 may be or contain a computer-readable medium, such as a floppy disk device, a hard disk device, an optical disk device, or a tape device, a flash memory or other similar solid state memory device, or an array of devices, including devices in a storage area network or other configurations. A computer program product can be tangibly embodied in an information carrier. The computer program product may also contain instructions that, when executed, perform one or more methods, such as those described above. The information carrier is a computer- or machine-readable medium, such as the memory 904, the storage device 906, or memory on processor 902.

[0066] The high speed controller 908 manages bandwidth-intensive operations for the computing device 900, while the low speed controller 912 manages lower bandwidth-

intensive operations. Such allocation of functions is exemplary only. In one implementation, the high-speed controller 908 is coupled to memory 904, display 916 (e.g., through a graphics processor or accelerator), and to high-speed expansion ports 910, which may accept various expansion cards (not shown). In the implementation, low-speed controller 912 is coupled to storage device 906 and low-speed expansion port 914. The low-speed expansion port, which may include various communication ports (e.g., USB, Bluetooth, Ethernet, wireless Ethernet) may be coupled to one or more input/output devices, such as a keyboard, a pointing device, a scanner, or a networking device such as a switch or router, e.g., through a network adapter.

[0067] The computing device 900 may be implemented in a number of different forms, as shown in the figure. For example, it may be implemented as a standard server 920, or multiple times in a group of such servers. It may also be implemented as part of a rack server system 924. In addition, it may be implemented in a personal computer such as a laptop computer 922. Alternatively, components from computing device 900 may be combined with other components in a mobile device (not shown), such as device 950. Each of such devices may contain one or more of computing device 900, 950, and an entire system may be made up of multiple computing devices 900, 950 communicating with each other.

[0068] Computing device 950 includes a processor 952, memory 964, an input/output device such as a display 954, a communication interface 966, and a transceiver 968, among other components. The device 950 may also be provided with a storage device, such as a microdrive or other device, to provide additional storage. Each of the components 950, 952, 964, 954, 966, and 968, are interconnected using various buses, and several of the components may be mounted on a common motherboard or in other manners as appropriate.

[0069] The processor 952 can execute instructions within the computing device 950, including instructions stored in the memory 964. The processor may be implemented as a chipset of chips that include separate and multiple analog and digital processing units. The processor may provide, for example, for coordination of the other components of the device

950, such as control of user interfaces, applications run by device 950, and wireless communication by device 950.

[0070] Processor 952 may communicate with a user through control interface 958 and display interface 956 coupled to a display 954. The display 954 may be, for example, a TFT LCD (Thin-Film-Transistor Liquid Crystal Display) or an OLED (Organic Light Emitting Diode) display, or other appropriate display technology. The display interface 956 may comprise appropriate circuitry for driving the display 954 to present graphical and other information to a user. The control interface 958 may receive commands from a user and convert them for submission to the processor 952. In addition, an external interface 962 may be provide in communication with processor 952, so as to enable near area communication of device 950 with other devices. External interface 962 may provide, for example, for wired communication in some implementations, or for wireless communication in other implementations, and multiple interfaces may also be used.

[0071] The memory 964 stores information within the computing device 950. The memory 964 can be implemented as one or more of a computer-readable medium or media, a volatile memory unit or units, or a non-volatile memory unit or units. Expansion memory 984 may also be provided and connected to device 950 through expansion interface 982, which may include, for example, a SIMM (Single In Line Memory Module) card interface. Such expansion memory 984 may provide extra storage space for device 950, or may also store applications or other information for device 950. Specifically, expansion memory 984 may include instructions to carry out or supplement the processes described above, and may include secure information also. Thus, for example, expansion memory 984 may act as a security module for device 950, and may be programmed with instructions that permit secure use of device 950. In addition, secure applications may be provided via the SIMM cards, along with additional information, such as placing the identifying information on the SIMM card in a non-hackable manner.

[0072] The memory may include, for example, flash memory and/or NVRAM memory, as discussed below. In one implementation, a computer program product is tangibly embodied in an information carrier. The computer program product contains instructions that, when executed, perform one or more methods, such as those described above. The information

carrier is a computer- or machine-readable medium, such as the memory 964, expansion memory 984, or memory on processor 952, that may be received, for example, over transceiver 968 or external interface 962.

[0073] Device 950 may communicate wirelessly through communication interface 966, which may include digital signal processing circuitry where necessary. Communication interface 966 may provide for communications under various modes or protocols, such as GSM voice calls, SMS, EMS, or MMS messaging, CDMA, TDMA, PDC, WCDMA, CDMA2000, or GPRS, among others. Such communication may occur, for example, through radio-frequency transceiver 968. In addition, short-range communication may occur, such as using a Bluetooth, WiFi, or other such transceiver (not shown). In addition, GPS (Global Positioning System) receiver module 980 may provide additional navigation- and location-related wireless data to device 950, which may be used as appropriate by applications running on device 950.

[0074] Device 950 may also communicate audibly using audio codec 960, which may receive spoken information from a user and convert it to usable digital information. Audio codec 960 may likewise generate audible sound for a user, such as through a speaker, e.g., in a handset of device 950. Such sound may include sound from voice telephone calls, may include recorded sound (e.g., voice messages, music files, etc.) and may also include sound generated by applications operating on device 950.

[0075] The computing device 950 may be implemented in a number of different forms, as shown in the figure. For example, it may be implemented as a cellular telephone 980. It may also be implemented as part of a smart phone 982, personal digital assistant, or other similar mobile device.

[0076] Various implementations of the systems and techniques described here can be realized in digital electronic circuitry, integrated circuitry, specially designed ASICs (application specific integrated circuits), computer hardware, firmware, software, and/or combinations thereof. These various implementations can include implementation in one or more computer programs that are executable and/or interpretable on a programmable system including at least one programmable processor, which may be special or general

purpose, coupled to receive data and instructions from, and to transmit data and instructions to, a storage system, at least one input device, and at least one output device.

[0077] These computer programs (also known as programs, software, software applications or code) include machine instructions for a programmable processor, and can be implemented in a high-level procedural and/or object-oriented programming language, and/or in assembly/machine language. As used herein, the terms “machine-readable medium” and “computer-readable medium” refer to any computer program product, apparatus and/or device (e.g., magnetic discs, optical disks, memory, Programmable Logic Devices (PLDs)) used to provide machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions as a machine-readable signal. The term “machine-readable signal” refers to any signal used to provide machine instructions and/or data to a programmable processor.

[0078] To provide for interaction with a user, the systems and techniques described here can be implemented on a computer having a display device (e.g., a CRT (cathode ray tube) or LCD (liquid crystal display) monitor) for displaying information to the user and a keyboard and a pointing device (e.g., a mouse or a trackball) by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback (e.g., visual feedback, auditory feedback, or tactile feedback); and input from the user can be received in any form, including acoustic, speech, or tactile input.

[0079] The systems and techniques described here can be implemented in a computing device that includes a back end component (e.g., as a data server), or that includes a middleware component (e.g., an application server), or that includes a front end component (e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the systems and techniques described here), or any combination of such back end, middleware, or front end components. The components of the system can be interconnected by any form or medium of digital data communication (e.g., a communication network). Examples of communication networks include a local area network (“LAN”), a wide area network (“WAN”), and the Internet.

[0080] The computing device can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

[0081] A number of embodiments have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention.

[0082] In addition, the logic flows depicted in the figures do not require the particular order shown, or sequential order, to achieve desirable results. In addition, other steps may be provided, or steps may be eliminated, from the described flows, and other components may be added to, or removed from, the described systems. Accordingly, other embodiments are within the scope of the following claims.

Claims

1. A computer-implemented method (1000) for editing a configuration of a physical entity to optimize a future state of the physical entity, the configuration including a plurality of discrete potential states for the physical entity wherein the potential states are described as values (SV1 to SV4) of a function (f1) of a base parameter (BP) on which the potential states depend, and wherein, for each given configuration, a set of inputs consistent with this configuration is given by a semi-analytic set described by explicit equality and inequality constraints on the inputs, the method comprising:

providing (1200), via a graphical user interface device (200), an initial visual representation of a default configuration wherein the plurality of potential states are represented by interactive graphical user interface objects (PS1 to PS4) which correspond to default function values at respective base parameter values, and providing at least one graphical result object (RO1) to a user (10) wherein the at least one graphical result object (RO1) represents at least one value characterizing the outcome of an optimization over a set of possible combinations of optimization inputs that are consistent with the default configuration and satisfy a pre-defined set of inequality constraints, the optimization being determined by an optimization algorithm (OA1) adapted to optimize the combination of inputs consistent with the default configuration under one or more given optimization constraints on the set of input combinations wherein the one or more optimization constraints are a further input to the optimization algorithm, and wherein the optimization algorithm has a component that defines the explicit equality and inequality constraints defining for any given configuration the semi-analytic set of inputs consistent with the given configuration;

receiving (1300), via a drag action (DA1) sensed by the user interface device (200), for at least one of the interactive graphical user interface objects (PS4) a corresponding user defined input to modify the associated default potential state into a user defined modified potential state as part of a user defined modified configuration;

re-applying (1400) the optimization algorithm (OA1) to the modified configuration and further optimizing the combination of inputs consistent with modified configuration under the optimization constraint; and

providing (1500) at least one modified graphical result object (RO2) to the user (10) wherein the at least one modified graphical result object represents at least one result value characterizing the outcome of the optimization over a set of combinations of inputs that are consistent with the modified configuration and satisfy the optimization constraints.

2. The method of claim 1, further comprising:

receiving (1100) a selection for the base parameter (BP) and the type (PST) of the potential states.

3. The method of claim 1 or 2, further comprising:

receiving (1600) a selection of a particular modified configuration or of the default configuration; and

deploying (1700) the selected configuration (c1) to the physical entity (30) to operate the physical entity in accordance with the selected configuration.

4. The method of any of the previous claims, wherein the optimization algorithm belongs to any one of the following optimization methods: standard gradient descent methods, dynamic programming methods, linear and nonlinear programming methods, or combinatorial optimization methods.

5. The method of any of the previous claims, further comprising:

receiving, via an insert action sensed by the user interface device (200), a corresponding user defined input for inserting a further graphical interactive user interface object as a further potential state of the modified configuration wherein the associated base parameter and state values are derived from the position of the inserted object in the graphical user interface;

and inserting the further graphical interactive user interface object as further potential state of the modified configuration.

6. The method of any of the previous claims, further comprising:

receiving, via a delete action sensed by the user interface device (200), a corresponding user defined input for deleting an existing graphical interactive user interface object, and

removing the existing interactive user interface object from modified configuration.

7. The method of any of the previous claims, wherein the outcome of the optimization satisfies user-specified constraints.

8. The method of any of the previous claims, wherein at least one value characterizing the outcome of an optimization is determined by a function of the combination of inputs.

9. A computer program product comprising instructions that, when loaded into a memory of a computing device and executed by at least one processor of the computing device, execute the method steps of the computer implemented method according to any one of the previous claims.

10. A computer system (100) for editing a configuration of a physical entity to optimize a future state of the physical entity, the configuration including a plurality of discrete potential states for the physical entity wherein the potential states are described as values (SV1 to SV4) of a function (f1) of a base parameter (BP) on which the potential states depend, and wherein, for each given configuration, a set of inputs consistent with this configuration is given by a semi-analytic set characterized by explicit equality and inequality constraints, the system comprising:

a graphical user interface component (200) configured:

to provide to the user (10) an initial visual representation of a default configuration wherein the plurality of potential states are represented by interactive graphical user interface objects (PS1 to PS4) which correspond to default function values at respective base parameter values, and to provide at

least one graphical result object (RO1) to the user wherein the at least one graphical result object (RO1) represents at least one corresponding result value characterizing the outcome of an optimization over a set of possible combinations of optimization inputs that are consistent and satisfy a pre-defined set of inequality constraints;

to sense a drag action (DA1) performed by the user on at least one of the interactive graphical user interface objects (PS4) as a user defined input to modify the associated default potential state into a user defined modified potential state as part of a modified configuration;

provide at least one modified graphical result object (RO2) to the user (10) wherein the at least one modified graphical result object represents at least one result value characterizing the outcome of an optimization over a set of combinations of inputs that are consistent with the modified configuration under the one or more given optimization constraints;

an optimizer component (120) communicatively coupled with the graphical user interface component (200), configured:

to determine the optimization for the default configuration by an optimization algorithm (OA1) adapted to optimize the combination of inputs consistent with the default configuration under the one or more given optimization constraints on the set of input combinations, wherein the optimization algorithm has a component that defines the explicit equality and inequality constraints defining for any given configuration the semi-analytic set of inputs consistent with the given configuration; and

to re-apply the optimization algorithm (OA1) to the modified configuration to further optimize the combination of inputs consistent with modified configuration under the one or more given optimization constraints.

11. The system of claim 10, wherein the graphical user interface component (200) is further configured:

to receiving a selection for the base parameter (BP) and the type (PST) of the potential states.

12. The system of claim 10 or 11, further configured to:

receive a selection of a particular modified configuration or of the default configuration; and

deploy the selected configuration (c1) to the physical entity (30) to operate the physical entity in accordance with the selected configuration.

13. The system of any of the claims 10 to 12, further configured to:

receive, via an insert action sensed by the user interface device (200), a corresponding user defined input for inserting a further graphical interactive user interface object as a further potential state of the modified configuration wherein the associated base parameter and state values are derived from the position of the inserted object in the graphical user interface;

and insert the further graphical interactive user interface object as further potential state of the modified configuration.

14. The system of any of the claims 10 to 13, further configured to:

receive, via a delete action sensed by the user interface device (200), a corresponding user defined input for deleting an existing graphical interactive user interface object, and

remove the existing interactive user interface object from modified configuration.

15. The method of any of the claims 10 to 14, wherein the outcome of the optimization satisfies user-specified constraints.

16. The system of any of the claims 9 to 15, wherein at least one value characterizing the outcome of an optimization is determined by a function of the combination of inputs.

Page 1/5

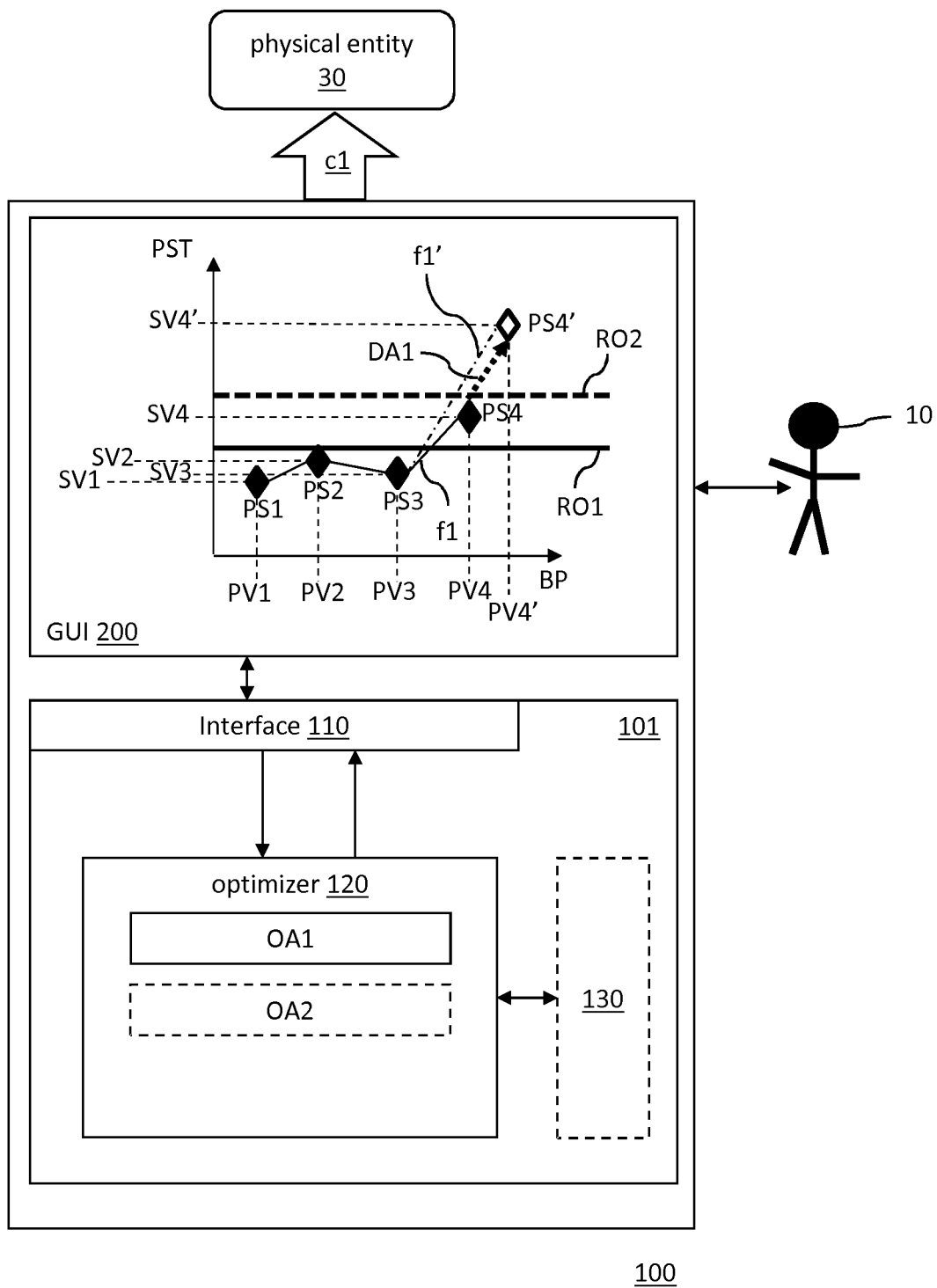


FIG. 1

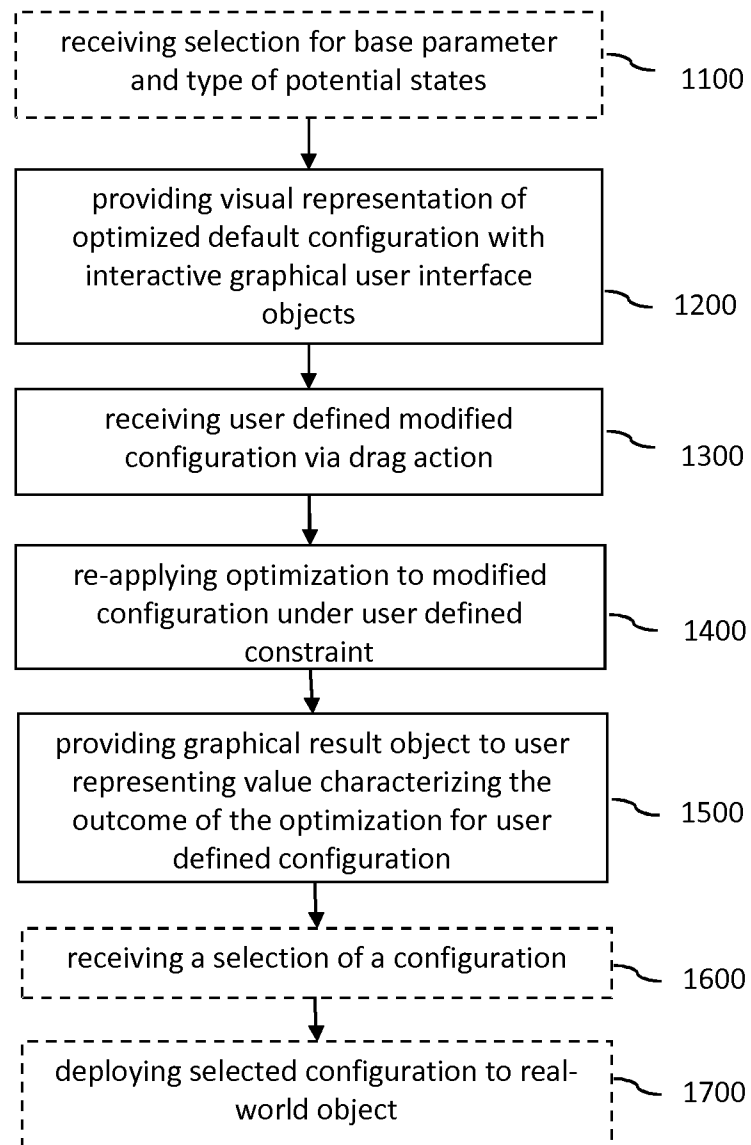
1000

FIG. 2

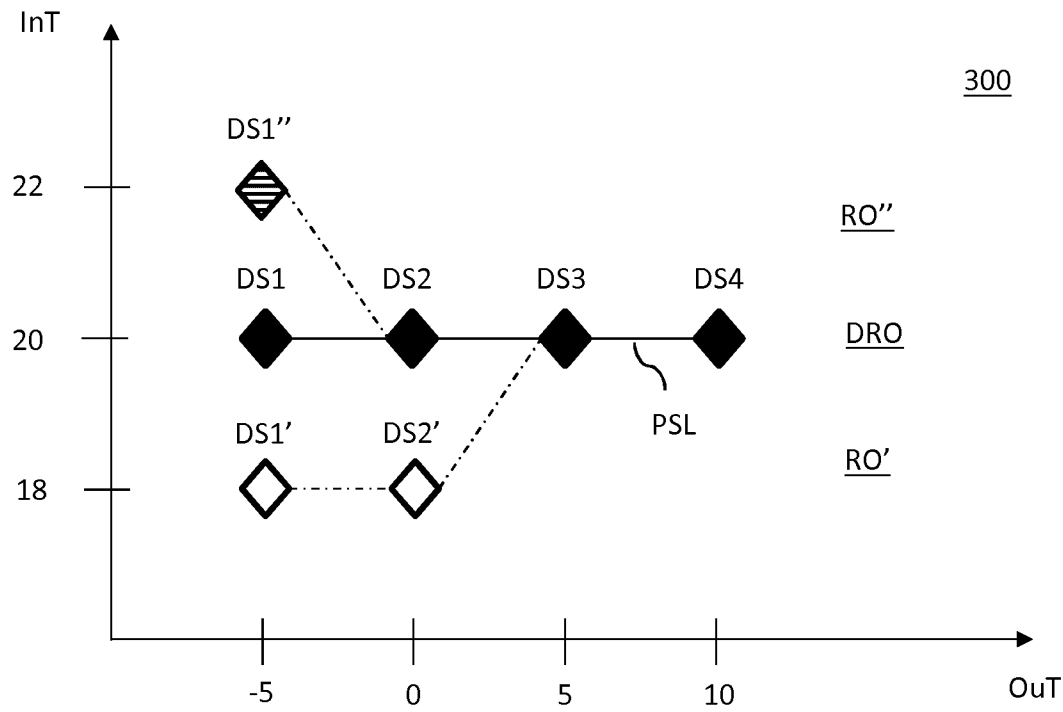


FIG. 3

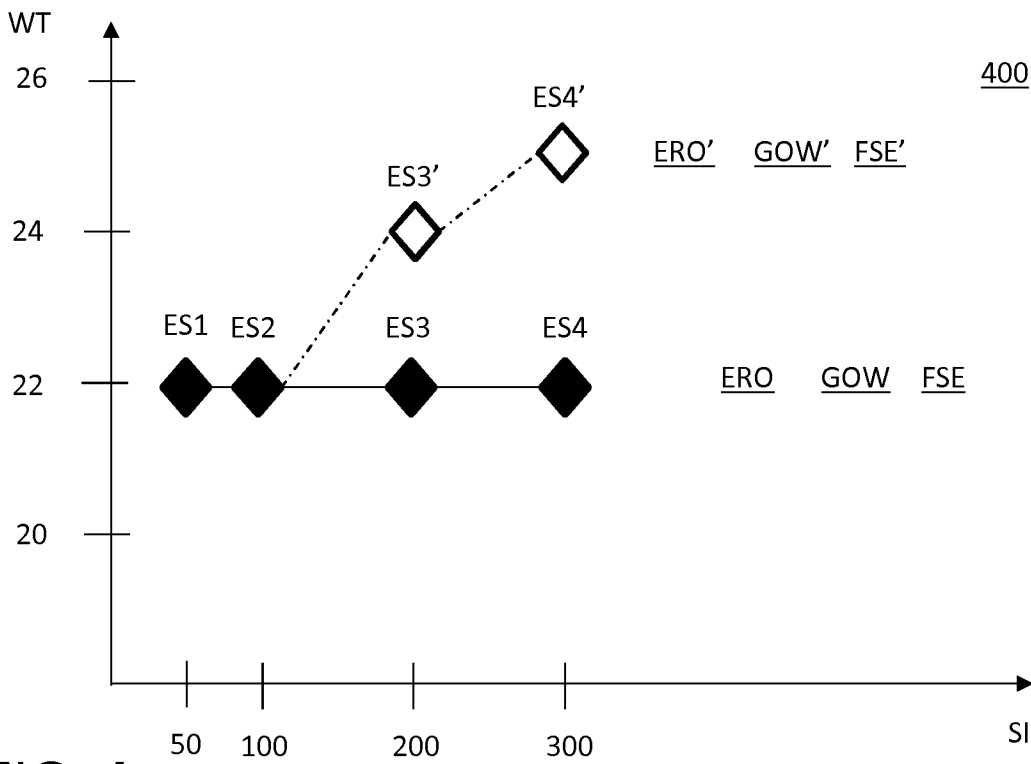


FIG. 4

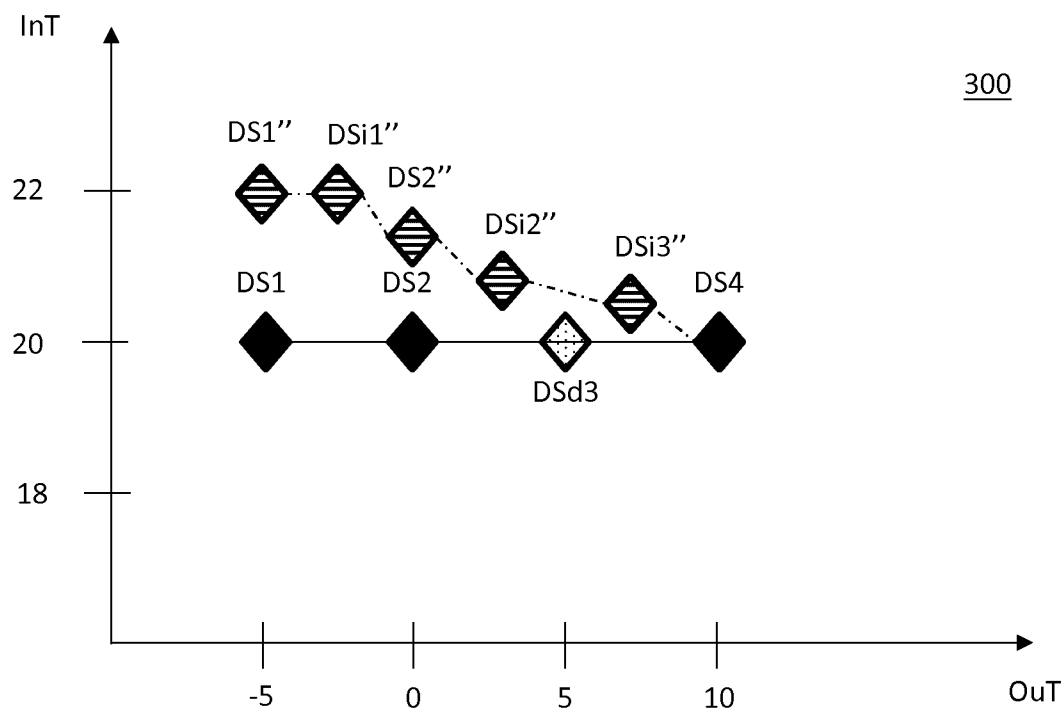


FIG. 5

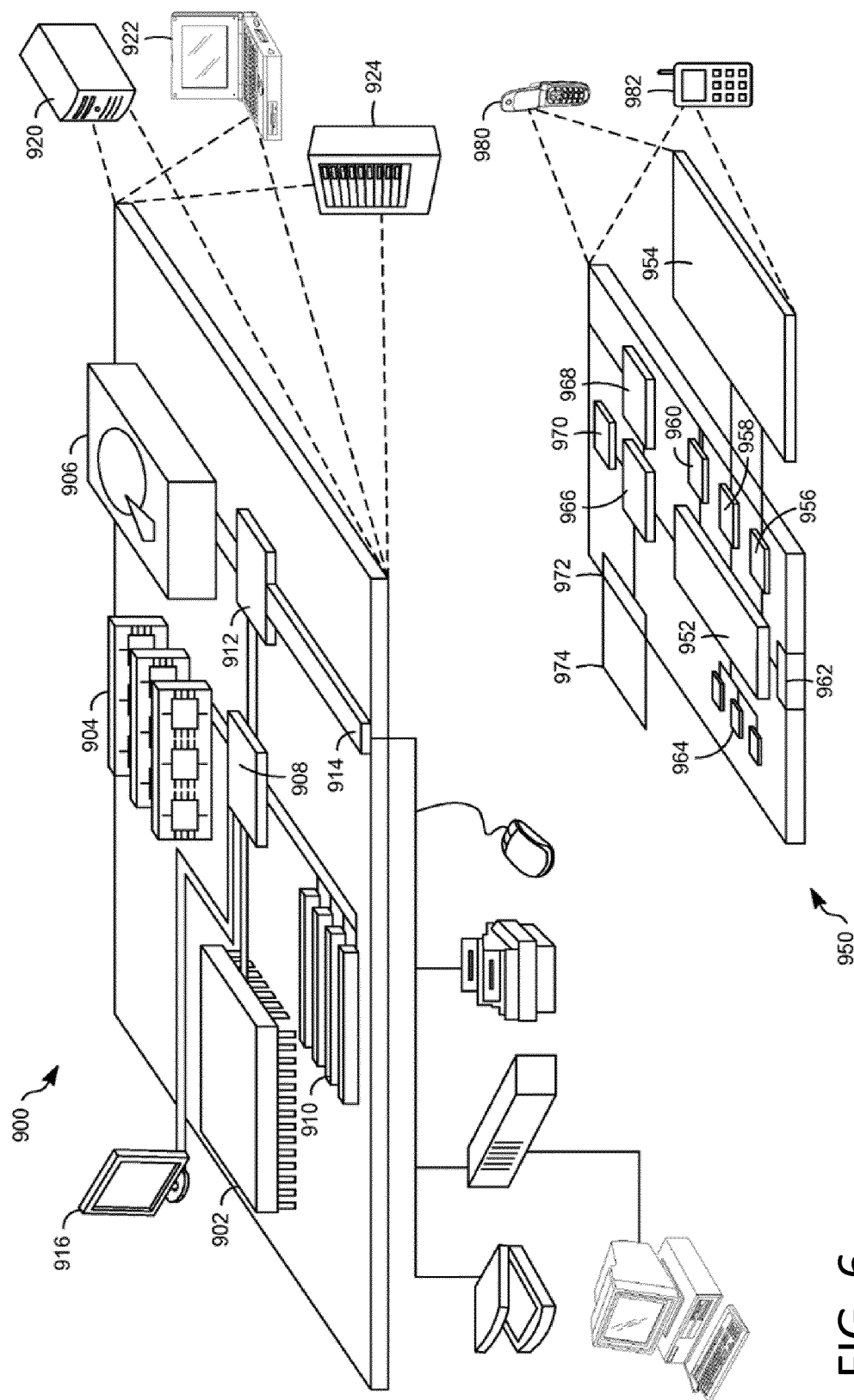


FIG. 6

PATENT COOPERATION TREATY

PCT

DECLARATION OF NON-ESTABLISHMENT OF INTERNATIONAL SEARCH REPORT

(PCT Article 17(2)(a), Rules 13ter.1(c) and Rule 39)

Applicant's or agent's file reference 2018P01252WO	IMPORTANT DECLARATION	Date of mailing(<i>day/month/year</i>) 24 July 2019 (24-07-2019)
International application No. PCT/EP2019/065002	International filing date(<i>day/month/year</i>) 7 June 2019 (07-06-2019)	(Earliest) Priority date(<i>day/month/year</i>) 18 January 2019 (18-01-2019)
International Patent Classification (IPC) or both national classification and IPC G06Q10/04 G05D23/19 G06Q50/06		
Applicant ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE (EPFL)		

This International Searching Authority hereby declares, according to Article 17(2)(a), that **no international search report will be established** on the international application for the reasons indicated below

1. ☐ The subject matter of the international application relates to:
 - a. ☐ scientific theories.
 - b. ☐ mathematical theories
 - c. ☐ plant varieties.
 - d. ☐ animal varieties.
 - e. ☐ essentially biological processes for the production of plants and animals, other than microbiological processes and the products of such processes.
 - f. ☐ schemes, rules or methods of doing business.
 - g. ☐ schemes, rules or methods of performing purely mental acts.
 - h. ☐ schemes, rules or methods of playing games.
 - i. ☐ methods for treatment of the human body by surgery or therapy.
 - j. ☐ methods for treatment of the animal body by surgery or therapy.
 - k. ☐ diagnostic methods practised on the human or animal body.
 - l. ☐ mere presentations of information.
 - m. ☐ computer programs for which this International Searching Authority is not equipped to search prior art.

2. ☒ The failure of the following parts of the international application to comply with prescribed requirements prevents a meaningful search from being carried out:

☐ the description
☒ the claims
☐ the drawings

3. ☐ The failure of the nucleotide and/or amino acid sequence listing to comply with the standard provided for in Annex C of the Administrative Instructions prevents a meaningful search from being carried out:

☐ the written form has not been furnished or does not comply with the standard.

☐ the computer readable form has not been furnished or does not comply with the standard.

4. Further comments:

Name and mailing address of the International Searching Authority  European Patent Office, P.B. 5818 Patentlaan 2 NL-2280 HV Rijswijk Tel. (+31-70) 340-2040 Fax: (+31-70) 340-3016	Authorized officer APIDIALI-MÜLLER, Sofia Tel: +49 (0)89 2399-8396
---	--

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 203

It is not possible for a skilled person to understand what kind of semi-analytic equalities and inequalities should be implemented in a generic case for an undefined physical entity and there is no clue as to what kind of equality or inequality could apply to both detailed examples / physical entities

- a heating system for a smart house with solar, wind and fossil energy,
- an indoor swimming pool heated with solar energy combined to a battery storing solar energy.

It is further noted that the claims are not restricted to heating/storing energy.

The claims fail to define a particular physical entity and fail to define constraints. It is not for the reader to fill the gaps, and the application is not sufficiently disclosed for a reader to select parameters and constraints, so that the requirements of Art. 5 PCT are not met.

The applicant's attention is drawn to the fact that claims relating to inventions in respect of which no international search report has been established need not be the subject of an international preliminary examination (Rule 66.1(e) PCT). The applicant is advised that the EPO policy when acting as an International Preliminary Examining Authority is normally not to carry out a preliminary examination on matter which has not been searched. This is the case irrespective of whether or not the claims are amended following receipt of the search report or during any Chapter II procedure. If the application proceeds into the regional phase before the EPO, the applicant is reminded that a search may be carried out during examination before the EPO (see EPO Guidelines C-IV, 7.2), should the problems which led to the Article 17(2) declaration be overcome.