(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 September 2012 (27.09.2012)

(10) International Publication Number
WO 2012/129123 A1

(51) International Patent Classification:
G09B 3/00 (2006.01)

(21) International Application Number:
PCT/US2012/029551

(22) International Filing Date:
16 March 2012 (16.03.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/052,623 21 March 2011 (21.03.2011) US

(71) Applicant *(for all designated States except US)*: **LEARN-BOP LLC** [US/US]; 4620 Henry Street, Pittsburgh, Pennsylvania 15213 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **TU, Arthur** [US/US]; 5706 Philips Avenue, Pittsburgh, Pennsylvania 15217 (US). **RAJAKUMAR, Bharanidharan** [US/US]; 1441 NW 19th Street, Apartment 109, Miami, Florida 33125 (US).

(74) Agent: **TAN, Carina, M.**; Reed Smith LLP, P.O. Box 488, Pittsburgh, Pennsylvania 15230 (US).

(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, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, 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, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report (Art. 21(3))*

(54) Title: LEARNING BEHAVIOR OPTIMIZATION PROTOCOL

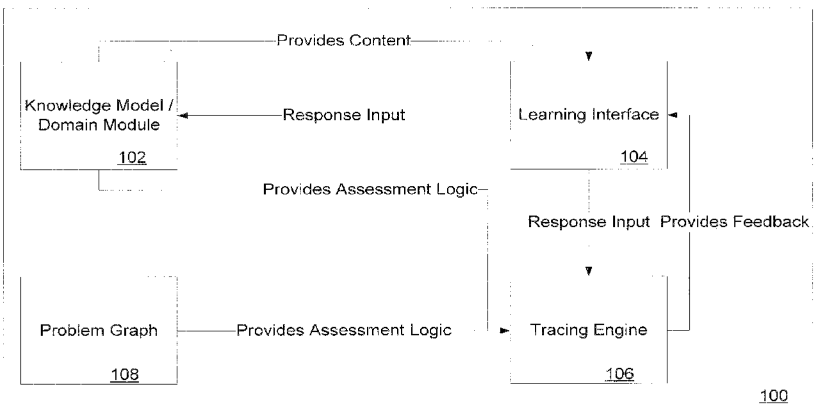

FIG. 1

(57) Abstract: According to certain aspects of some embodiments, LearnBop is both a conceptual and a logical design for a two-way, reciprocating learning platform and community where users can create, consume, critique, review learning progress and improve learning content.

WO 2012/129123 A1

Learning Behavior Optimization Protocol (LearnBop)

TECHNICAL FIELD

[0001] The present invention is directed to online learning, and more specifically to an interactive and adaptive learning environment with knowledge-centered componentization.

BACKGROUND

[0002] FIG. 1 is a general data flow diagram of a conventional intelligent tutoring system. FIG. 1 shows tutoring system 100 comprising knowledge model/domain module 102, learning interface 104, problem graph 108 and tracing engine 106.

[0003] While such construction poses a good analogy of a "knowledgeable" human tutor that reacts to students' queries with appropriate conceptual feedback, such a concept of a tutoring system results in a complex and often disorderly design of the tracing engine 106 that needs to communicate with both the domain module 102 and the learning interface 104, which often have no standardized one-to-one mapping between knowledge and user interactions. There are a number of downsides to this conventional concept of intelligent tutoring systems since they are: 1) Difficult to organize, due to the knowledge and interface modules' independence from each other. Any change made to the knowledge module will not automatically produce relevant interface components, and vice versa; 2) Inefficient to execute and operate, as the knowledge module runs a separate process from the interface module in processing student input. Independent knowledge and interface modules communicating in parallel may depend on allocation of additional computer and networking resources (e.g., thread, communication port); 3) Non-modular - with separate modules and processes, it is difficult to take a particular intelligent tutoring problem and extract and recombine constituent knowledge and steps; 4) Non-reusable - since neither the knowledge module nor the interface module fully define the conceptual entirety of the problem, the intelligent tutor may not be easily ported to other platforms (e.g., smart phones, tablets, kiosks, e-readers, portable gaming consoles) without rewriting one or more of the knowledge module, the interface module or the tracing engine in order to re-define the relationships between knowledge and interface interactions.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] FIG. 1 is a general data flow diagram of a conventional intelligent tutoring system.

[0005] FIG. 2 is a high-level data flow diagram that illustrates an overview of the CKALE paradigm and the design of the LearnBop platform, according to certain embodiments.

[0006] FIG. 3 illustrates the design architecture of an interaction knowledge component, according to certain embodiments.

[0007] FIG. 4 illustrates a high-level logical design of a lesson on the LearnBop platform, according to certain embodiments.

[0008] FIG. 5 is an example of rich-text content authored for use in a LearnBop adaptive lesson, according to certain embodiments.

[0009] FIG. 6 illustrates an instructional scaffolding example, according to certain embodiments.

[0010] FIG. 7 illustrates another instructional scaffolding example, according to certain embodiments.

[0011] FIG. 8 illustrates creating a representation using the authoring process, according to certain embodiments.

[0012] FIG. 9 illustrates conceptual labeling in the authoring process, according to certain embodiments.

[0013] FIG. 10 illustrates designation of interaction knowledge components, according to certain embodiments.

[0014] FIG. 11 illustrates populating Interaction knowledge components, according to certain embodiments.

[0015] FIG. 12 illustrates the rendering of a given representation into an adaptive lesson, according to certain embodiments.

[0016] FIG. 13 illustrates an example of a hint request button, according to certain embodiments.

[0017] FIG. 14 illustrates the use of a modal message, according to certain embodiments.

[0018] FIG. 15 illustrates a progress display, according to certain embodiments.

[0019] FIG. 16 illustrates instructional scaffolding, according to certain embodiments.

[0020] FIG. 17 illustrates the use of focus grabbers, according to certain embodiments.

[0021] FIG. 18 illustrates Focus-Sensitive Problem-Solving Step #1, according to certain embodiments.

[0022] FIG. 19 illustrates Focus-Sensitive Problem-Solving Step #2, according to certain embodiments.

[0023] FIG. 20 is a graph illustrating the amount of time each user/learner spent on the lesson, according to certain embodiments.

[0024] FIG. 21 illustrates a Conditional and Correlational Analysis Example – Hint Effectiveness, according to certain embodiments.

[0025] FIG. 22 illustrates a sample Motivation and Strategy for Learning Questionnaire, according to certain embodiments.

[0026] FIG. 23 illustrates a Help-Seeking Behavior Reporting Example – Hints Requests vs. Intrinsic Motivation, according to certain embodiments.

[0027] FIG. 24 illustrates Predicting Future Help Needs – Decision Tree, according to certain embodiments.

[0028] FIG. 25 illustrates a sample Causal Model of Learning, Motivation and Help-Seeking, according to certain embodiments.

[0029] FIG. 26 shows data flow of the system 2600 per interaction knowledge component, according to certain embodiments.

[0030] FIG. 27 illustrates a Service-based Client Design, according to certain embodiments.

[0031] FIG. 28 illustrates an Offline Client Design, according to certain embodiments.

DETAILED DESCRIPTION

[0032] <u>A New Adaptive Learning Environment With Knowledge-Centered Componentization</u>

[0033] According to certain embodiments, Learning Behavior Optimization Protocol (LearnBop) is a componentized learner-, knowledge- and skill- centered, motivationally- and metacognitively-enhanced learning platform design that allows explanation-driven, representation-sensitive and context-sensitive authoring to create learning content for use on personal computers, mobile devices as well as on devices without network connectivity. According to certain aspects of some embodiments, LearnBop is both a conceptual and a logical design for a two-way, reciprocating learning platform and community where users can create, consume, critique, review learning progress and improve learning content.

[0034] Componentized, knowledge-centered adaptive learning environment (CKALE), is a method for creating responsive knowledge that adapts to incorrect or correct responses, according to certain embodiments. CKALE has been implemented as the LearnBop Platform.

[0035] FIG. 2 is a high-level data flow diagram that illustrates an overview of the CKALE paradigm and the design of the LearnBop platform, according to certain embodiments.

[0036] FIG. 2 illustrates the LearnBop platform as a learning environment 200 constructed by small building blocks called interaction knowledge components 202 and includes problem flow control 206 and messaging control 204, according to certain embodiments. Interaction knowledge components 202 resemble both conceptual and software sub-components of a learning exercise. The interaction knowledge component 202 is an independent, severable unit of instruction and learning interaction that can provide feedback to students through messaging control 204, or can be chained together to form more complex problems with problem flow control 206. Interaction knowledge component 202 includes input interface 208, assessment logic 212 and knowledge definition 210.

[0037] In short, traditional intelligent tutoring systems are divided into holistic modules and exchange information between modules in holistic manners. For instance, a problem graph defines correct inputs based on states of the interface. Thus, a simple input like "5" as an addition operand at different times may have different prior states. Similarly, a domain module may use certain rules and logic to evaluate certain fields of the interface, and even though two fields on the interface demonstrate the same skill, the rules in the domain module need to be distinctly bound (or hook) to every input field. Many concerns like the ones mentioned here that arise from interactions of holistic modules in traditional intelligent tutoring systems, make problem authoring extremely difficult to generalize and authored problems hard to reuse within and across different platforms.

[0038] In contrast, the CKALE methodology takes on the requirements of generalized, flexible problem authoring and reusable authored problems more easily: every interaction knowledge component contains compact interface manifestation and assessment logic to represent the evaluation of knowledge in the form of a single input. Therefore, no software interface component on the screen is without a direct mapping to associated knowledge. Such a design allows the same concepts and skills in a learning problem to be reused by simply adding an interaction knowledge component and without having to create additional bindings or hooks between the interface and domain modules or problem graphs. Furthermore, as long as a new platform implements the set of LearnBop interaction knowledge components, a problem authored on the LearnBop platform can be reproduced on the new platform without explicit modifications.

[0039] The CKALE paradigm sets a new standard for adaptive learning where learning environments and learning systems are constructed with complete coherence to the conceptual construction of the topic of instruction, as opposed to traditional intelligent tutoring systems, where software systems function and interface with domain modules as separate processes.

[0040] The CKALE paradigm and the design of the LearnBop platform comprise the following:

[0041] **Interaction knowledge components**, are compact reusable, regroupable modules that fully define the relevant domain knowledge (e.g., what is the coefficient of a term 3x), the visual manifestation of the knowledge on the interface (e.g., a problem prompt complemented with a textbox input), as well as all control logic to evaluate correctness and provide instructional scaffolding. Therefore, an interaction knowledge component as a modular encapsulation of knowledge serves as a fundamental building block to complex problem solving and problem authoring, allowing one to divide or combine problems and study learning content in part, in whole or in conglomerates.

[0042] **Behavior Optimization Protocol Definition Language, or BOP definition language (BDL)**, is a high level mark-up language used to initialize, order, chain and populate interaction knowledge components in order to fully define learning interactions in adaptive lessons. Since interaction knowledge components may have slightly different implementations on different platforms (e.g., desktop computers vs. tablets), the BDL serves as an important underlying foundation to lesson generation since it provides a standardized way of describing interactions, making the same adaptive lesson reusable across different media without explicit modification.

[0043] **Learning Environment Interface** is a generic visual environment that houses the interfaces and interactions produced by CKALE. The learning environment interface assumes several requirements, including means to request hint, movable windows, attention grabbers and modal window locks.

[0044] **Knowledge-centered, representation-sensitive authoring process**, is one that uses a What-You-See-Is-What-You-Get (WYSIWYG) style visual manipulation tool to create adaptive lessons without requiring the user to explicitly create BDL definitions. The authoring process emphasizes visual manifestation of superset and subset relations. In other words, interaction knowledge components may be dropped into a color-coded concept container, and will then be treated as a conceptual whole that the platform will present and scaffold holistically.

[0045] **Knowledge Discovery as a Service (KDS)** provides state-of-the-art analysis and reporting services from the learning sciences, to any instructors that deploy classes on a CKALE system. Instead of presenting bare statistics and reports on the raw data (e.g., raw student inputs, correctness of answers) generated by the system, CKALE system performs machine learning on all the learning behaviors that took place on the learning platform, and present instructors highly refined models that predict student performance and learning style, so as to help the instructor discover specific learning patterns .

[0046] **Adaptive Learning as a Service (ALS)**, is an cloud-computing metaphor for education where through distributed computing apparatuses, BDL-defined adaptive lessons can turn web services (and distributed computing as well as local computing apparatuses alike) into learning resource and instructional scaffolding providers. A wide range of devices with or without network connectivity can deliver full-fledged adaptive learning experience to learners in a wide-range of developed and underdeveloped social and infrastructural settings, supplying true ubiquitous learning.

[0047] The CKALE paradigm is one where computerized knowledge can be divided, joined, regrouped and reused effectively and efficiently, allowing for adaptive learning over a wide range of networks and computing devices.

[0048] The following sections contain the designs for each of the constituent modules of the LearnBop system: Interaction knowledge components, Bop Definition Language, Authoring Process, Knowledge Discovery and Adaptive Learning as a Service.

[0049] Interaction Knowledge Components

[0050] An interaction knowledge component is a fundamental building block in a componentized, knowledge-centered adaptive learning environment that resembles both a sub-concept/sub-skill resulting from cognitive task analysis in the learning sciences, and a software design architecture.

[0051] FIG. 3 illustrates the design architecture of an interaction knowledge component, according to certain embodiments.

[0052] As illustrated in FIG. 3, interaction knowledge component 300 includes an input interface 304, an assessment logic 302, and a knowledge definition 306.

[0053] Input interface 304 is a visual manifestation of the interaction knowledge component that provides the user with a prompt (video, audio or other media) and software interface components (textboxes, radio buttons, drop-down lists, drag-and-drop lists or other interface elements).

[0054] Assessment logic 302 is responsible for evaluating user input. Interaction knowledge component 300 may have multiple correct answers; for each correct answer there may be a different success feedback message; for each incorrect answer there may be a different error message; each interaction knowledge component may also provide a variable number of hints that the learner can request.

[0055] Knowledge definition 306 provides the content that will populate the prompt and input controls on the interface, and to the assessment logic to evaluate correctness of inputs.

[0056] The input interface and the assessment logic provide an abstract, reusable building block for interactive knowledge representation that is later populated by specific knowledge definitions.

[0057] Concept Grouping and Tagging

[0058] As described previously, an interaction knowledge component is capable of evaluating a granular conceptual or skill step such as adding, subtracting or citing a fact (the list is by no means exhaustive). However, many more complex skills such as derivation, integration, tracing graph tours, calculating conditional probabilities, may require multiple granular steps to complete.

[0059] Therefore, it is often beneficial to organize interaction knowledge components into concepts that describe complex skills. The role of concepts is similar to the interaction knowledge component. It is a building block that can be reused and regrouped both in conceptual grounds and in software engineering.

[0060] The following is an example of a complex skill organized by a concept:

[0061] $3(2+5) = 6 + 15$

[0062] As shown above, the skill of integer multiplication involves distributing the 3 and multiplying it by 2 and 5 respectively. In other words, the skill described here requires two interaction knowledge components to demonstrate. Hence, the individual interaction knowledge components and overall concept in this example are tagged accordingly.

[0063] Lesson Exercise Sequence

[0064] A typical practice problem in learning and in education often involves multiple concepts. FIG. 4 illustrates the logical design 400 of a lesson on the LearnBop platform (number of items shown in the diagram does not resemble any physical limitation of the system), according to certain embodiments.

[0065] FIG. 4 shows Interaction knowledge component chaining and problem formation (lesson exercise 402). FIG. 4 shows that interaction knowledge components 406 may function as independent incremental steps in a problem, but they can be chained together either into one problem, or into multiple concepts 404 that form one problem/lesson exercise 402.

[0066] <u>Behavior Optimization Protocol Definition Language</u>

[0067] As described previously, interaction knowledge components may be grouped to form concepts and exercises. In order to achieve such a degree of reusability and flexibility, the CKALE paradigm includes the use of a generalized definition language to specify the content of a lesson.

[0068] A plausible but not the only implementation of such a definition language is XML that can be used to define a lesson exercise, like so:

[0069] (Example 1)

[0070] <Lesson>

[0071] <Exercise>

[0072] <Name>Calculus Practice</Name>

[0073] <DefaultHintMessage>Default Hint for the entire lesson.</DefaultHintMessage>

[0074] <DefaultErrorMessage>Default Error for the entire lesson.</DefaultErrorMessage>

[0075] <Concept>

[0076] <Name>Arithmetic</Name>

[0077] <DefaultHintMessage>Default Hint for the concept: Arithmetic</DefaultHintMessage>

[0078] <DefaultErrorMessage>Default Error for the concept: Arithmetic</DefaultErrorMessage>

[0079] <Component>

[0080] <TextBox>

[0081] <Prompt>Enter the missing value</Prompt>

[0082] <HintMessage>Add the integers together.</HintMessage>

[0083] <HintMessage>What is 3+2?</HintMessage>

[0084] <Input>

[0085] <Value>5</Value>

[0086] <IsCorrect>true</IsCorrect>

[0087] <SuccessMessage>Good Job!</SuccessMessage>

[0088] </Input>

[0089] <DefaultErrorMessage>Add the integers 3 and 2.</DefaultErrorMessage>

[0090] </TextBox>

[0091] </Component>

[0092] </Concept>

[0093] <Concept>

[0094] <Name>Derivatives</Name>

[0095] <DefaultHintMessage>Default Hint for the concept: Derivatives</DefaultHintMessage>

[0096] <Component>

[0097] <TextBox>

[0098] <Prompt>Enter the missing value</Prompt>

[0099] <HintMessage>To derive an algebraic expression, multiply the exponent by the coefficient and subtract the exponent by one.</HintMessage>

[00100] <Input>

[00101] <Value>15</Value>

[00102] <IsCorrect>true</IsCorrect>

[00103] <SuccessMessage>Good Job!</SuccessMessage>

[00104] </Input>

[00105] <Input>

[00106] <Value>10</Value>

[00107] <IsCorrect>false</IsCorrect>

[00108] <ErrorMessage>This is incorrect, multiply 5 by 3 here.</ErrorMessage>

[00109] </Input>

[00110] <ErrorMessageDefault>This is incorrect.</ErrorMessageDefault>

[00111] </TextBox>

[00112] </Component>

[00113] <Component>

[00114] <MultipleChoice>

[00115] <Prompt>Enter the missing value</Prompt>

[00116] <HintMessage>When deriving an algebraic expression, subtract 1 from the exponent</HintMessage>

[00117] <Input>

[00118] <Value>2</Value>

[00119] <IsCorrect>true</IsCorrect>

[00120] <SuccessMessage>Good Job!</SuccessMessage>

[00121] </Input>

[00122] <Input>

[00123] <Value>3</Value>

[00124] <IsCorrect>false</IsCorrect>

[00125] <ErrorMessage>You forgot to subtract the exponent, 3, by 1. </ErrorMessage>

[00126] </Input>

[00127] <ErrorMessageDefault>This is incorrect.</ErrorMessageDefault>

[00128] </MultipleChoice>

[00129] </Component>

[00130] </Concept>

[00131] </Exercise>

[00132] </Lesson>

[00133] The above markup outlines a definition written in a XML-based implementation of the BOP definition language. The example outlines a small derivative problem with two sets of interaction knowledge components, the first set demonstrating the concept and skill of arithmetic operations, and the second set demonstrates the concept of derivatives. As can be seen in the example, the lesson may contain exercises, which in turn contain concepts and interaction knowledge components. One can define the default hint and error message at each of the different hierarchical levels described above. One can also create hint messages for every interaction knowledge component, or error messages for every input value that is handled by the interaction knowledge component. The example also contains definitions for two types of inputs, textboxes and multiple choices.

[00134] The example is by no means exhaustive of the possibilities, as it is presented to demonstrate the immense flexibility and reusability suggested by the CKALE

paradigm where concepts and skills in learning can be regrouped, joined and divided.

[00135] Authoring Process

[00136] According to certain embodiments, techniques are provided for generating an adaptive lesson, which adaptive lesson is constructed by an instructor without requiring any knowledge of computer programming and only requiring access to the internet. The instructor-developed lessons are learner and knowledge specific and fully specify the conceptual or skill-based knowledge points that a learner must focus on via interaction knowledge components.

[00137] Lesson Content Authoring

[00138] Using the authoring process, instructors can create a complete lesson tailored to a student's specific interests. The curriculum designer scaffolds curriculum design for the instructor in a step-by-step manner.

[00139] Rich-text and Multimedia Content

[00140] In numerous cases, the instructor authoring the bop may wish to include static lesson content for students to consume before starting an exercise. The LearnBop platform authoring process therefore includes a WYSIWYG (What you see is what you get) editor for creating rich-text and multimedia lesson content that can be included in and deployed as a part of an adaptive lesson for delivering a fuller learning experience. FIG. 5 is an example of rich-text content authored for use in a LearnBop adaptive lesson, according to certain embodiments.

[00141] FIG. 5 shows rich-text content example 500 that illustrates definition 502 of "slope" on a curve, derivative equation 504 and explanation 506.

[00142] Instructional Scaffolding

[00143] In accessing static content, instructional scaffolding that provides additional context-specific learning content upon request are often beneficial to learning. The LearnBop platform lesson content authoring tool provides means for an instructor to highlight part of the rich-text multimedia content and provide additional, optional scaffolding information for target learners.

[00144] For example, FIG. 6 illustrates an instructional scaffolding example, according to certain embodiments. FIG. 6 shows a lesson snapshot 600 with the definition 604 of the term "derivative" 602 inserted and requested.

[00145] FIG. 7 illustrates another instructional scaffolding example, according to certain embodiments. FIG. 7 shows a lesson snapshot 700 with an additional scaffolding message 704 to help the learner understand a new way of looking (702) at a problem.

[00146] Visual Authoring Process

[00147] Once the curriculum design process has collected sufficient information to guide the lesson authoring process, a visual authoring process is initiated to help teachers rapidly create adaptive lessons without programming or design work.

[00148] As discussed previously, an interaction knowledge component is a modular component that encapsulates interface components necessary to demonstrate and manifest a concept visually (e.g., radio buttons and a submit button for multiple choice), as well as associated conceptual knowledge (e.g., hints, error messages, prompts, etc) required to scaffold a student to successfully complete the problem or recover from errors. Thus, an interaction knowledge component not only acts as a building block for the interface, it is also a representation of a step in a problem-solving process. A concept on the other hand subsumes one or more interaction knowledge components to illustrate a more complex concept or skill in learning.

[00149] In order to allow creation of complex adaptive lessons without design or programming experience, the LearnBop architecture incorporates a visual authoring process in the CKALE paradigm to serve as guidelines for authoring tools for adaptive learning problems.

[00150] The CKALE authoring process design is grounded in a WYSIWYG (what you see is what you get) interface where visual representations can be manipulated by dragging-and-dropping, and information may be inputted through the keyboard.

[00151] In overview, the adaptive authoring process is divided into a number of phases as follows, according to certain embodiments:

[00152] **Creating a representation**

[00153] By using visual tools such as ink-based/touch-based drawing, graphics manipulation, equation editors, etc, an instructor/author can create a representation that illustrates the content of the problem. FIG. 8 illustrates creating a representation using the authoring process, according to certain embodiments.

[00154] For example, to create a derivative problem that resembles what was described in the BOP definition language section, an author/instructor may create a representation 800 as shown in FIG. 8.

[00155] <u>Labeling concepts</u>

[00156] An instructor can use one or more resizable, color-coded labels to select part of the presentation that demonstrate particular concepts or skills. FIG. 9 illustrates conceptual labeling in the authoring process, according to certain embodiments.

[00157] To illustrate, following the example used previously in FIG. 8, an instructor/author may label concepts using color-coded blocks 902 for representation 900 in FIG. 9.

[00158] <u>Designating interaction knowledge components</u>

[00159] FIG. 10 illustrates designation of interaction knowledge components, according to certain embodiments. The author/instructor may use resizable, color-coded labels 1002 to designate interaction knowledge components, transforming the representation 1000 into an adaptive problem.

[00160] <u>Populating interaction knowledge components</u>

[00161] FIG. 11 illustrates populating Interaction knowledge components, according to certain embodiments. The author/instructor may use visual forms and other common user interface controls to populate information for the interaction knowledge component that have been added to the lesson.

[00162] For example, FIG. 11 shows that the author may populate the hint messages by adding the messages 1102 to the list 1100.

[00163] <u>Publishing the visual content into BOP definition language</u>

[00164] Upon completing the creation of the lesson, the author may publish the lesson to BOP definition language. The process of publishing is straightforward. Since the ownership hierarchy of exercises, concepts and interaction knowledge components are explicitly illustrated by the visual manifestations of the lessons, the

implemented publishing process can quickly transform such visual hierarchy into one described in a BOP definition language. In addition, the visual manifestation also explicitly contains information required to crop the images necessary for deployment of the lessons. Finally, the representation stored in BOP definition language, as shown in the sample markup language explained above in the BOP Definition Language section, will be rendered into a learning interface. FIG. 12 illustrates the rendering of a given representation into an adaptive lesson, according to certain embodiments.

[00165] FIG. 12 shows the very same lesson 1200 after rendering and is ready for answer input 1206 in view of the coefficients 1202, 1204 of the equations shown in lesson 1200. The rendered screen bears high resemblance of the authoring screen.

[00166] Learning Environment Interface

[00167] Hint Request Button

[00168] Similar to scaffolding in the lesson content, when a learner is engaged in a learning exercise, it will be helpful to provide hints on the current step that the learner is working on. FIG. 13 illustrates an example of a hint request button, according to certain embodiments. The LearnBop platform design and the CKALE paradigm includes the use of one or more "hint" button 1302 that the user/learner can interact with to request additional scaffolding on the current exercise 1300, as shown in FIG. 13.

[00169] Modal Messaging

[00170] While engaged in a learning exercise, the user/learner may request a hint, may commit an error or may even require further instructions on using the interface. Therefore, a modal message box that locks the interface until the user explicitly closes the box is required to deliver information critical to the learning process. FIG. 14 illustrates the use of a modal message, according to certain embodiments. For example, FIG. 14 shows a modal message box 1402 displaying a hint for the learner.

[00171] Progress Display

[00172] The LearnBop platform design and the CKALE paradigm include the use of a visual manifestation of learning progress to inform the user of goal achievements. FIG. 15 illustrates a progress display, according to certain embodiments. FIG. 15 illustrates a non-limiting example of an implementation of a learning progress display shown as a progress bar 1502 for exercise 1500.

[00173] Instructional Scaffolding

[00174] As described previously, additional scaffolding information may be added to certain parts of the lesson. FIG. 16 illustrates instructional scaffolding, according to certain embodiments. This feature is available during adaptive exercises as well. FIG. 16 shows that exercise 1600 includes additional scaffolding information 1604 that is rendered when a learner requests additional information through button 1602.

[00175] Focus Grabber

[00176] Traditional intelligent tutoring systems' interfaces are often populated by numerous software interface components, and thus are rather overwhelming for the learner to process upon first arrival. From a learning science perspective, the overload of visual information consumes more cognitive resources, leaving less memory and attention for the user to focus on the exercise / learning task. Therefore, the LearnBop platform design and the CKALE paradigm include the use of "focus grabbers" to bring learners' attention to interface components that are important to the current step in the learning process. FIG. 17 illustrates the use of focus grabbers, according to certain embodiments.

[00177] FIG. 17 shows a non-limiting implementation of a "focus grabber" in the form of a blinking arrow 1702 in exercise 1700.

[00178] Focus-sensitive problem-solving

[00179] Like the "focus grabber", it is understood that learning is more effective when cognitive resources like attention and memory are fully allocated toward the learning task. As mentioned previously, traditional intelligent tutoring systems often have interfaces with large numbers of active interface controls.

[00180] Therefore, the LearnBop platform design and the CKALE paradigm implements a learning environment that includes at least one step-wise mechanism to divide the

problem into conceptual sub-components, and reveal only what is necessary for the current step in order to avoid distracting and overloading the learner with too much information. FIG. 18 illustrates Focus-Sensitive Problem-Solving Step #1, according to certain embodiments.

[00181] FIG. 18 show a non-limiting implementation of a step-wise problem-solving mechanism where the first concept/skill 1802 and its constituent steps are shown.

[00182] After the first concept or skill (in this case, writing out the derivative) has been completed, the system will then reveal the second concept or skill, in order to bring the learner's focus to the new sub-component of the exercise. FIG. 19 illustrates Focus-Sensitive Problem-Solving Step #2, according to certain embodiments. In FIG. 19, second concept or skill 1902 is brought to the learner's attention.

[00183] <u>Knowledge Discovery as a Service (Data Mining and Machine Learning)</u>

[00184] The LearnBop platform records the following types of log events on learning, with timestamps and user identifiers:

- Page Actions (a user enters or interacts with the interface)
- Activation (a user clicks on a interaction knowledge component, bringing focus to the component)
- Inputs (a user inputs a value as responses to the interface)
- Help request (a hint or additional instructional scaffolding information was requested by the student)
- Message displayed (a message, for instance, an error message, has been displayed on the interface)
- Close Window (a user closes a window or a popup on the interface)
- Answer is correct (the system determined the response input to be correct)
- Answer is incorrect (the system determined the response input to be incorrect)
- Completion (when an exercise has been fully completed)

[00185] The raw data provides a means to discover a number of ways to understand learning.

[00186] <u>Basic statistics</u>

[00187] The immediate benefit of the data is the basic statistics that include the amount of time spent on a lesson, number of hints requested, number of errors committed, as etc. The LearnBop platform provides the capability to produce aggregates, averages and other attributes of the aforementioned log events. The examples given in this section are by no means exhaustive.

[00188] FIG. 20 is a graph illustrating the amount of time each user/learner (notated as blue squares 2002) spent on the lesson (in seconds) , according to certain embodiments.

[00189] Learning Tracing: Conditional and Correlational Analysis

[00190] In addition to aggregates and averages of log events, the LearnBop platform also has the capacity to produce reports on conditional measures such as the effectiveness of hint messages (i.e., success rate on interaction knowledge components conditioned on hint requests), as well as correlational analysis such as success rate vs. time to help teachers understand whether students are investing meaningful study time or are they simply stuck. The LearnBop platform offers the capacity to compute conditional measures and conduct correlational analysis on aggregates, averages and other attributes of the log events in order to provide more detailed feedback on student learning. FIG. 21 illustrates a Conditional and Correlational Analysis Example – Hint Effectiveness, according to certain embodiments.

[00191] FIG. 21 shows a visualization 2100 of a step-wise problem in a lesson, and the reported success rate 2102 of response attempts after particular hints have been requested on the step.

[00192] Motivational and Meta-cognitive Measures

[00193] The LearnBop platform is an adaptive learning platform with an emphasis on learning science, which means the LearnBop platform augments the learning data collected with information regarding a student's meta-cognition and motivation, therefore providing possibilities of predicting future learning, something that has been extremely difficult to do in the past using just data on student performance.

[00194] The LearnBop platform augments the learning data reports with meta-cognitive and motivational information in the following ways:

[00195] Survey

[00196] In the learning science literature, surveys such as Motivation and Strategy for Learning Questionnaire (MSLQ) have been widely used to collect student self-reported measures on goal orientation, task value, intrinsic motivation, help-seeking behavior and other motivational and meta-cognitive constructs, by means of Likert scales.

[00197] FIG. 22 illustrates a sample Motivation and Strategy for Learning Questionnaire. In FIG. 22, the MSLQ questionnaire 2200 measures Extrinsic Goal Orientation 2202.

[00198] Such survey responses may be used to create new aggregates, averages or attributes for statistical analysis mentioned previously.

[00199] Help Seeking

[00200] As mentioned previously, the LearnBop platform provides a number of means for students to get help, including requests for hints, glossary term definitions and additional instructional scaffolding information. The usage information on all these scaffolding facilities reveal important information about students' meta-cognitive behaviors that may be used to understand how to improve students' future learning.

[00201] For example, if a student continuously enters incorrect answers and never asked for hints or other forms of help, this student is understood to be lacking in help-seeking behaviors.

[00202] Another example would be if a student consistently asks for all the hints, or inputting large numbers of answers within short periods of time, the student can be understood to be gaming the system.

[00203] Similarly, help seeking observations may also be used to create new aggregates, averages or attributes for statistical analysis mentioned previously.

[00204] FIG. 23 illustrates a Help-Seeking Behavior Reporting Example – Hints Requests vs. Intrinsic Motivation, according to certain embodiments. FIG. 23 is a visualization of how a motivational measure such as intrinsic motivation 2302, may relate to help-seeking behavior like the number of hints requested 2304. This type of visualization is invaluable to teachers who wish to understand how they might be

able to intervene inside or outside of class to increase student interest and strategy use in learning activities.

[00205] Predictions

[00206] Predictions are made possible by performing machine learning algorithms on the aggregates, averages and attributes of log events as well as higher-level motivational and meta-cognitive constructs mentioned previously.

[00207] The following prediction models have been incorporated into the LearnBop platform design:

[00208] Predicting better future learning

[00209] By employing machine learning algorithms such as Bayesian classification, Artificial Neural Networks and other viable alternatives, prediction models based on features from learning data and motivational/meta-cognitive constructs such as time spent on lesson and help-seeking behavior, are developed to predict student performance or for skill mastery.

[00210] FIG. 24 illustrates Predicting Future Help Needs – Decision Tree, according to certain embodiments. FIG. 24 shows a visualization of a learning optimization/prediction model implemented as a decision tree 2400 where depending on what steps 2402 of the problem the student answers correctly 2404 or incorrectly 2406, the model will recommend additional hints 2408 or suggest that student try an easier problem.

[00211] Causal Search

[00212] Another interesting class of machine learning algorithms is causal model search algorithms like PC, FCI, GES, LINGAM. By performing causal model search on the aggregates, averages and attributes mentioned previously, the LearnBop system can create causal models that estimate causal relationships between different measures and constructs.

[00213] For instance, if we have three measures/constructs such as performance, goal orientation, and time on lesson, there are many different causal models that may arise. One possible model may be that the students' goal orientation will affect how much effort they put in, which will be manifested as time on lesson and

performance. Therefore in this model, goal orientation is likely to be the cause of time on lesson and performance. However it may also be the case that students' time on lesson and performance affects their goal orientation in that if students are able to complete the lessons correctly in a short amount of time, they may set a goal to complete the lesson. Therefore in the second model, both time on lesson and performance are likely to be causes of goal orientation.

[00214] By providing visualized causal models, the LearnBop platform is providing in-depth analysis of learning that unveil insights to how teachers may be able to assist students both in electronic and in physical settings.

[00215] FIG. 25 illustrates a sample Causal Model of Learning, Motivation and Help-Seeking, according to certain embodiments. FIG. 25 shows a visualization of a causal model 2500.

[00216] As shown in FIG. 25, some relationships such as the ones between Mastery 2502 and Performance 2504, and between Self-Efficacy 2506 and Performance 2504, the direction of causation have been determined. For the other relationships that cannot be determined by causal search algorithms, the visualization will at least indicate whether the two constructs or variables are positively or negatively correlated.

[00217] Adaptive Learning as a Service (Ubiquitous Learning)

[00218] Another important feature of LearnBop's flexibility is that it allows users/authors to create and deploy the adaptive lessons once, and allows access to the same learning environment everywhere, whether it is on personal computers, on mobile devices or on offline devices without network connectivity.

[00219] A knowledge definition written in BOP definition language is created on the server, along with necessary resource files (e.g., images, audio, video, etc) to deliver a full adaptive lesson. There are three types of clients that can be developed and used to access the adaptive lessons created on LearnBop:

[00220] Web-based (browser-based) client

[00221] The web-based/browser-based client is the default LearnBop client that can be accessed by any device with network connectivity and an up-to-date web browser. The web-based client offers pre-compiled learning interfaces for each adaptive

lesson, full logging service for all learning behaviors and complete learning reports with visualizations. FIG 26 illustrates a Web-based/Browser-based client design, according to certain embodiments. FIG. 26 shows data flow of the system 2600 per interaction knowledge component. System 2600 includes a browser-based input interface 2602, an assessment logic 2604, a knowledge definitions library 2606, a logging control 2608 and database storage 2610, according to certain embodiments.

[00222] Service-based (mobile device) client

[00223] Some mobile devices may not have browsers that support modern scripting (e.g. AJAX) and style sheet (e.g. CSS) technologies, required to use the web-based client. The alternative is to use a service-based client.

[00224] According to certain embodiments, the LearnBop platform comes with a web service that provides the following services.

[00225] Authentication

[00226] This service authenticates the user and grants access to the subsequent services.

[00227] Lesson Search

[00228] (Requires authentication) This service returns a list of lessons that match certain search requirements (e.g., keywords, rating).

[00229] Adaptive Learning

[00230] (Requires authentication) Once the user enters an adaptive lesson, the client can connect to the rest of the web service to request information on interaction knowledge components, submit responses to interaction knowledge components and receive responses regarding whether or not the submitted responses were correct.

[00231] Logging

[00232] (Passive) Since the web service is hosted as a part of the LearnBop platform, all learning behaviors that were observable by the web service will be logged. The client does not actively control logging.

[00233] Data Reports

[00234] (Requires authentication) The user may retrieve statistics on learning from the web service.

[00235] In overview, the service-based architecture of the LearnBop platform provides mobile devices without proper browsers the freedom to implementation visual manifestations of interaction knowledge components (for instance, the service-based client need to provide interface components for multiple choice), and still have access to all the adaptive learning content and associated resources (e.g., images, videos, audios) like the traditional web-based clients.

[00236] In other words, for devices that do not have adequate browser support, the LearnBop platform web services will provide all the information necessary to create a customized third party client for learning and for reporting. FIG. 27 illustrates a Service-based Client Design, according to certain embodiments. FIG. 27 shows data flow per interaction knowledge component for a service-based client. System 2700 includes web service 2702, assessment logic 2704, knowledge definitions library 2706, database storage 2708, logging control 2710 and mobile devices 2712.

[00237] Offline client

[00238] Under some circumstances, users may not have access to devices with network connectivity. For such situations, the LearnBop platform offers a utility to generate a standalone lesson package for one adaptive lesson that can be accessed by a Javascript and CSS-enabled browser. Since without network connectivity, content changes to the adaptive lesson will not be reflected in the standalone package, learning behaviors will not be logged to the server and thus learning reports will not be available to the user. Therefore, the use of the offline client is strong discouraged.

[00239] The standalone lesson package does, however, include a local logging utility that generates a log file that can be manually retrieved and uploaded to the server at a later date. FIG. 28 illustrates an Offline Client Design, according to certain embodiments.

[00240] FIG. 28 shoes the data flow per interaction knowledge component for an offline client. System 2800 of FIG. 28 includes offline client generator 2802, browser based interface 2804, assessment logic 2806, knowledge definitions library 2808, standalone package 2810, database storage 2812, log import utility 2814, local logging 2816 and devices 2818 without connectivity.

We Claim:

1. A computer-implemented method of providing a platform for creating online interactive learning experience, the method comprising:

providing a plurality of computer-implemented interaction knowledge components as building blocks to build a plurality of concepts, one or more of which can be scaffolded to build an online interactive and adaptive lesson, wherein a respective interaction knowledge component of the plurality of interaction knowledge components is an independent severable unit of instruction and includes assessment logic and wherein the plurality of interaction knowledge components and concepts are re-usable and regroupable to build different online interactive and adaptive lessons.

2. The method of Claim 1, wherein a respective interaction knowledge component further comprises an input interface for providing a user one or more audio, video or other media prompts and interface components comprising one or more of: textboxes, drop-down lists, radio buttons, and drag-and-drop lists.

3. The method of Claim 1, wherein the assessment logic evaluates correctness of user input and provides feedback message based on the user input.

4. The method of Claim 2, wherein a respective interaction knowledge component further comprises a knowledge definition component that provides content for populating prompt and input controls on the input interface.

5. The method of Claim 1, further comprises using a high level mark up definition language to initialize, order, chain and populate interaction knowledge components.

6. The method of Claim 1, wherein a respective concept can be divided and into its respective interaction knowledge components and regrouped with other interaction knowledge components for reuse. 7. The method of Claim 1, further comprises providing a computer-implemented online automated curriculum designer to allow an instructor to create the online interactive and adaptive lesson for one or more users, wherein the online automated curriculum designer provides feedback to the instructor based on concept data aggregated from at least a subset of adaptive lessons created by a plurality of instructors, the feedback comprising identification of missing concepts or insufficiency of concepts of the adaptive lesson that the instructor is creating.

8. The method of Claim 1, further comprises using a computer–implemented visual authoring tool to allow creation of complex adaptive lessons without requiring design or programming experience.

9. The method of Claim 8, further comprises using computer-implemented conceptual labeling including color-coded labels to select a portion of a presentation to demonstrate a concept or skill.

10. The method of Claim 8, further comprises using computer-implemented interaction component designation for transforming a created representation into an adaptive problem for a respective user.

11. The method of Claim 1, further comprises using computer-implemented visual forms and user interface controls for populating information associated with the respective interaction knowledge components.

12. The method of Claim 1, further comprises publishing hierarchical visual content using a high level mark up definition language.

13. The method of Claim 1, further comprises implementing a hint button for allowing a user to interact with the adaptive lesson by requesting a hint for solving a problem in the adaptive lesson.

14. The method of Claim 1, further comprises providing graphical modal messages when a user requires further hints to solve a problem in the adaptive lesson.

15. The method of Claim 1, further comprises providing a visual display indicating a user's learning progress.

16. The method of Claim 1, further comprises using computer-implemented graphical focus grabbers to bring a user's attention to interface components that are important in the learning process.

17. The method of Claim 1, further providing a logical curriculum designer for:

allowing lesson content scaffolding using rich-text and multimedia content; and

allowing instructional scaffolding that provides context-specific learning content and messages to the us.

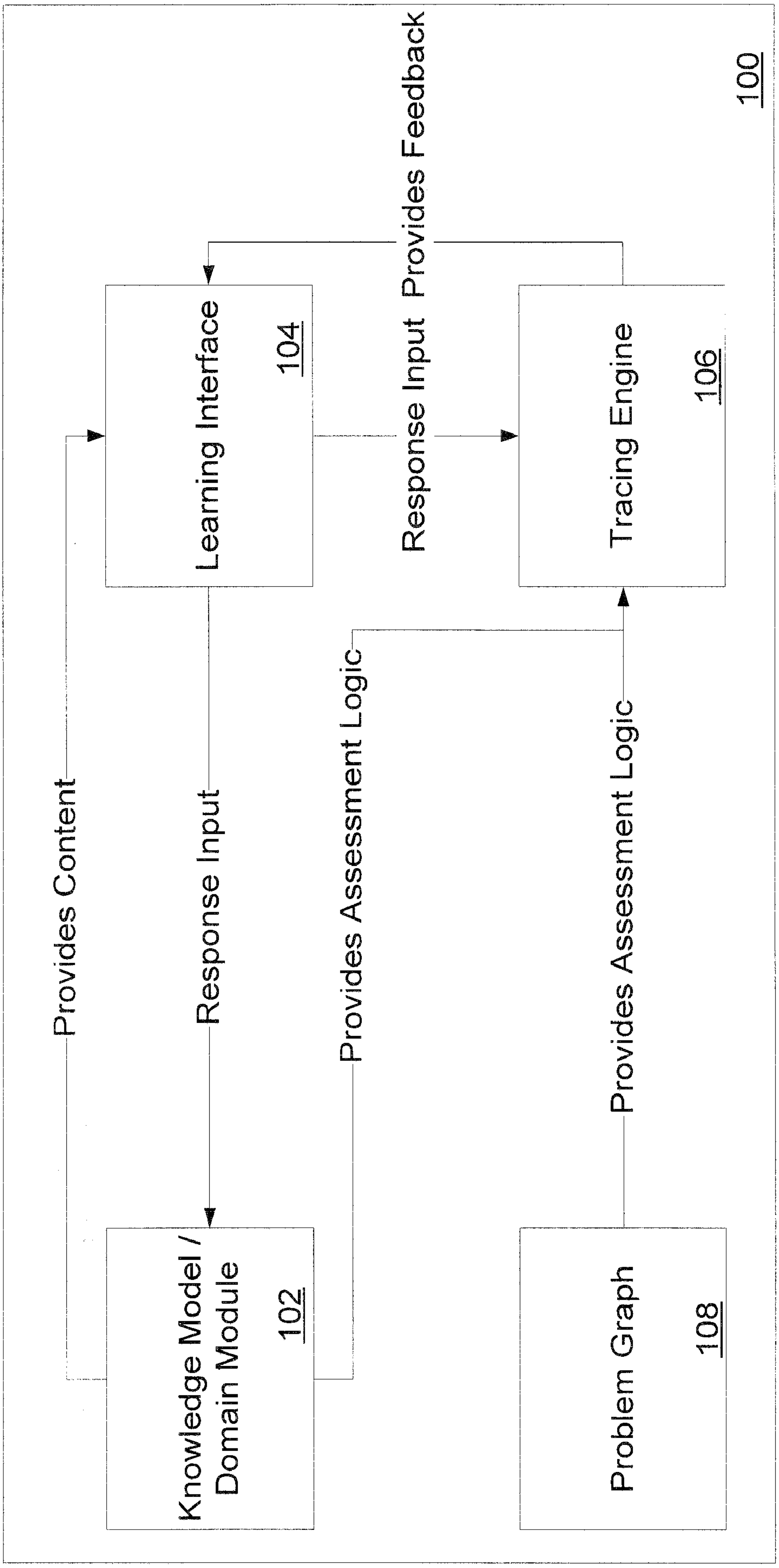
100
Learning Interface
104
Tracing Engine
106
Knowledge Model / Domain Module
102
Problem Graph
108
Provides Feedback
Response Input
Provides Content
Response Input
Provides Assessment Logic
Provides Assessment Logic

FIG. 1

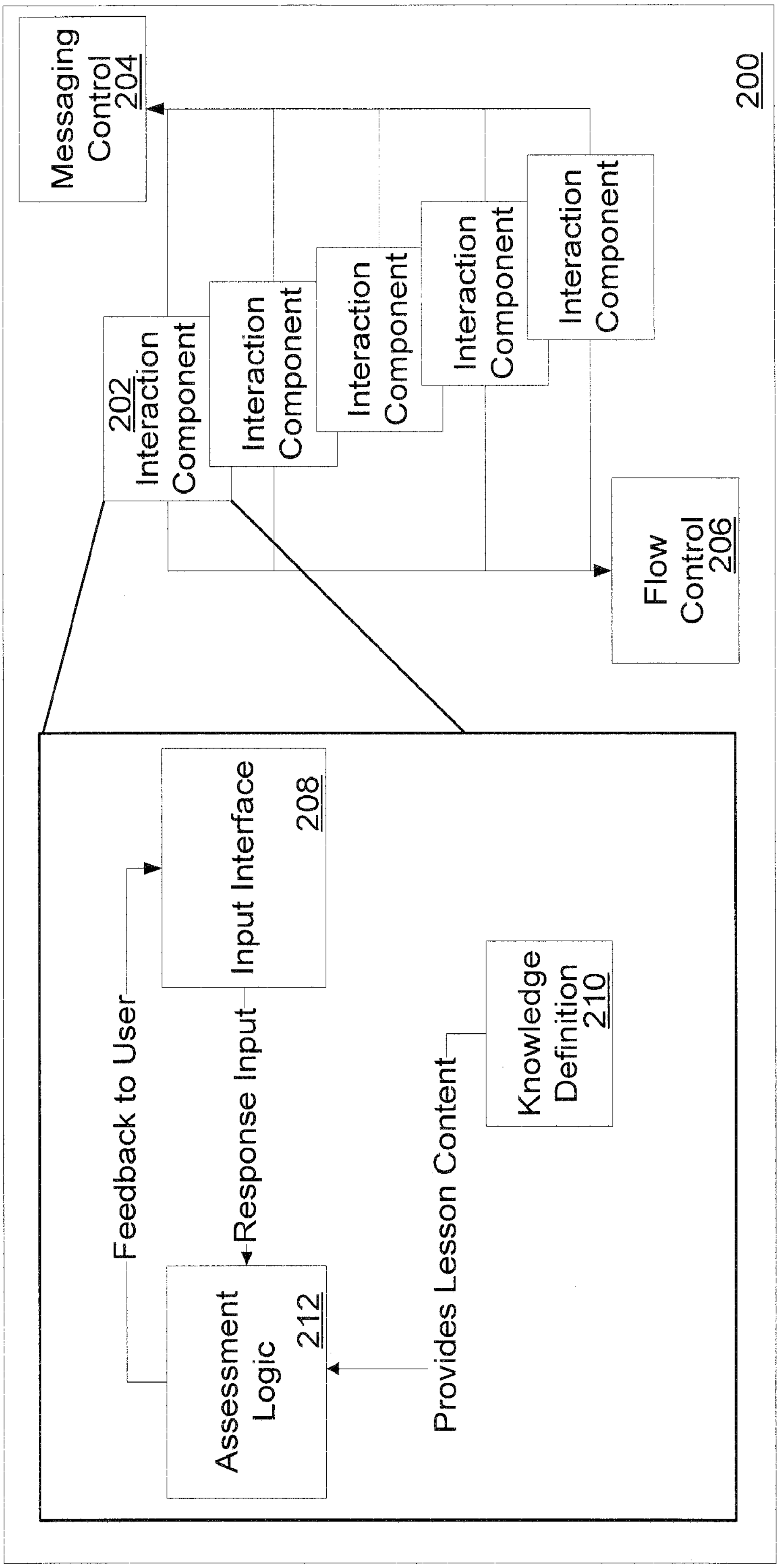
Messaging Control 204
202 Interaction Component
Interaction Component
Interaction Component
Interaction Component
Interaction Component
Flow Control 206
200
Input Interface 208
Feedback to User
Response Input
Provides Lesson Content
Knowledge Definition 210
Assessment Logic 212

FIG. 2

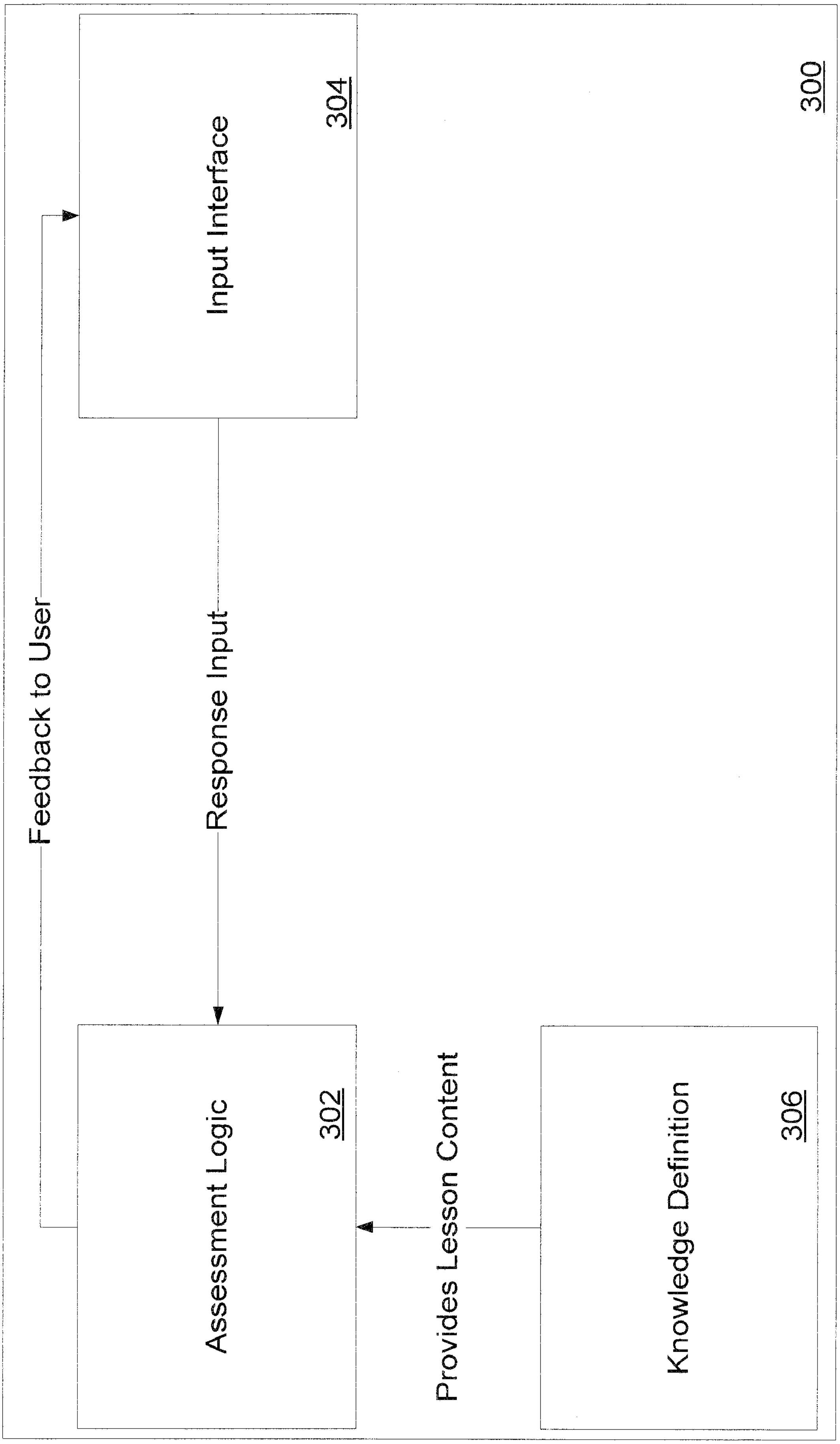
300
Input Interface
304
Feedback to User
Response Input
Assessment Logic
302
Provides Lesson Content
Knowledge Definition
306

FIG. 3

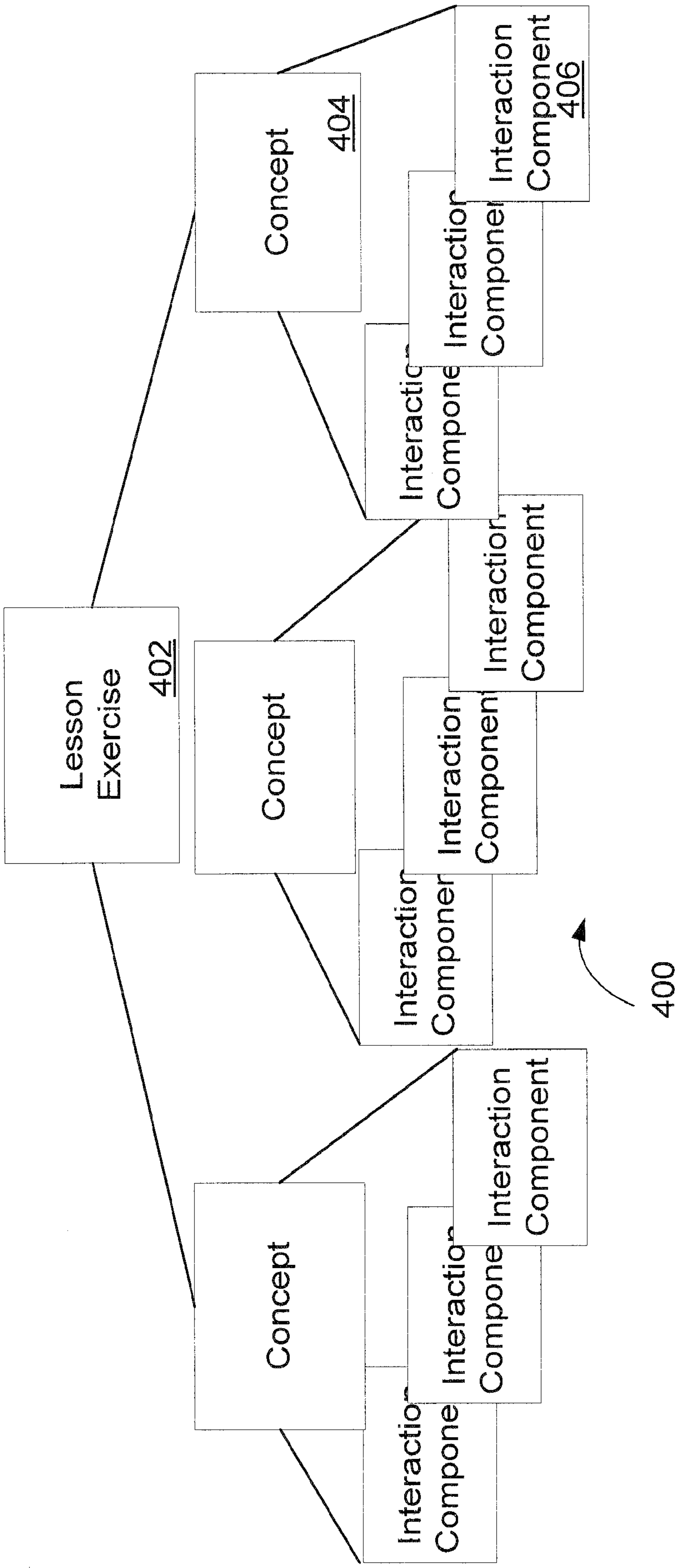
Lesson
Exercise
402
Concept
404
Interaction
Component
406
Concept
Interaction
Component
Interaction
Component
Concept
Interaction
Component
Interaction
Component
400

FIG. 4

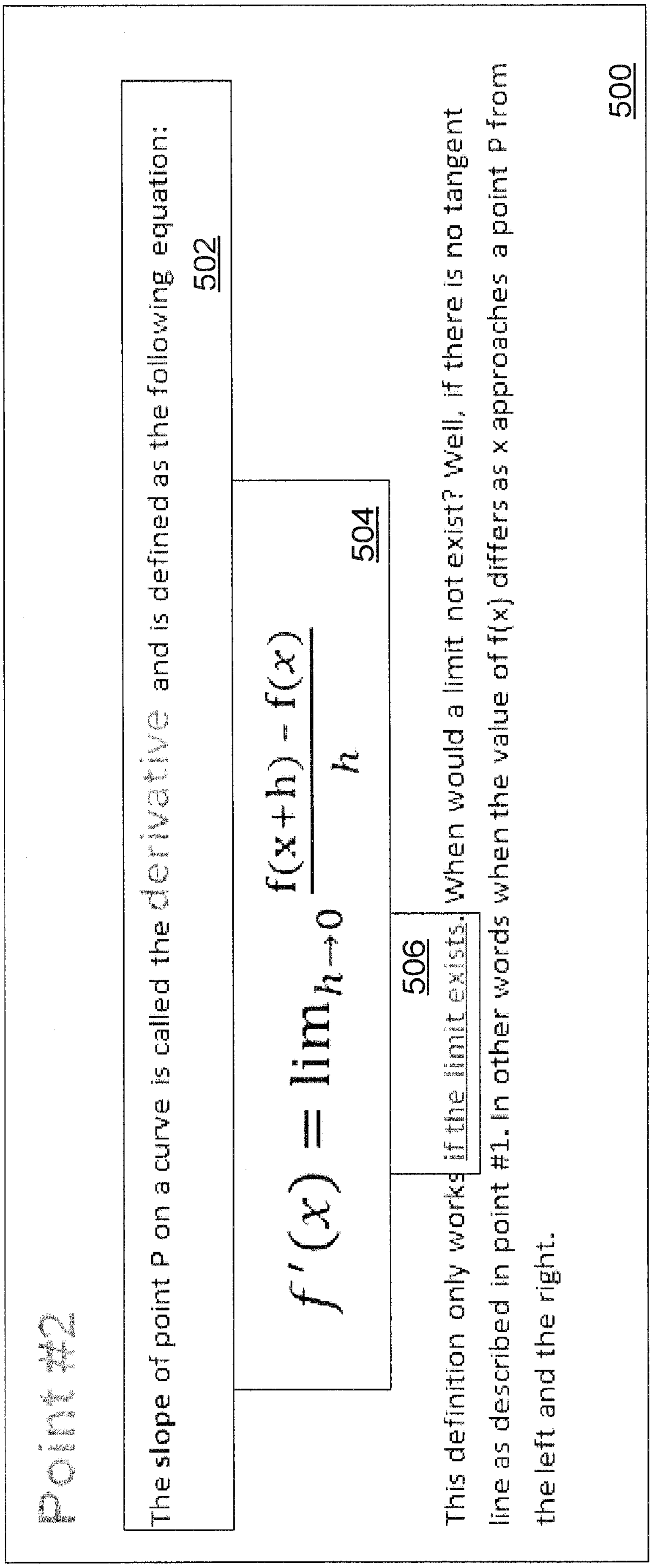
Point #2
The slope of point P on a curve is called the derivative and is defined as the following equation:
502
$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$
504
506
This definition only works if the limit exists. When would a limit not exist? Well, if there is no tangent line as described in point #1. In other words when the value of f(x) differs as x approaches a point P from the left and the right.
500

FIG. 5

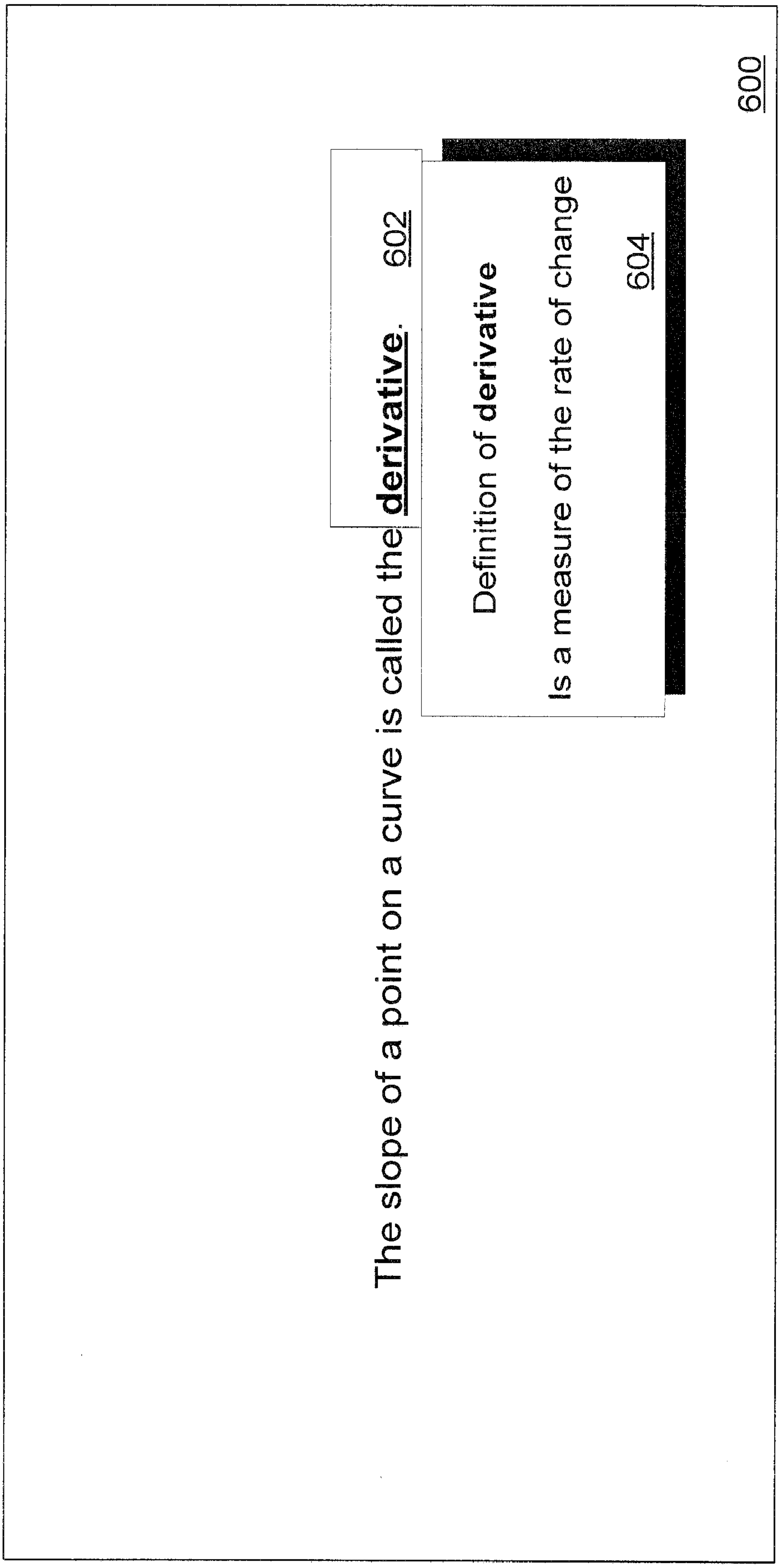
600
The slope of a point on a curve is called the derivative.
602
Definition of derivative
Is a measure of the rate of change
604

FIG. 6

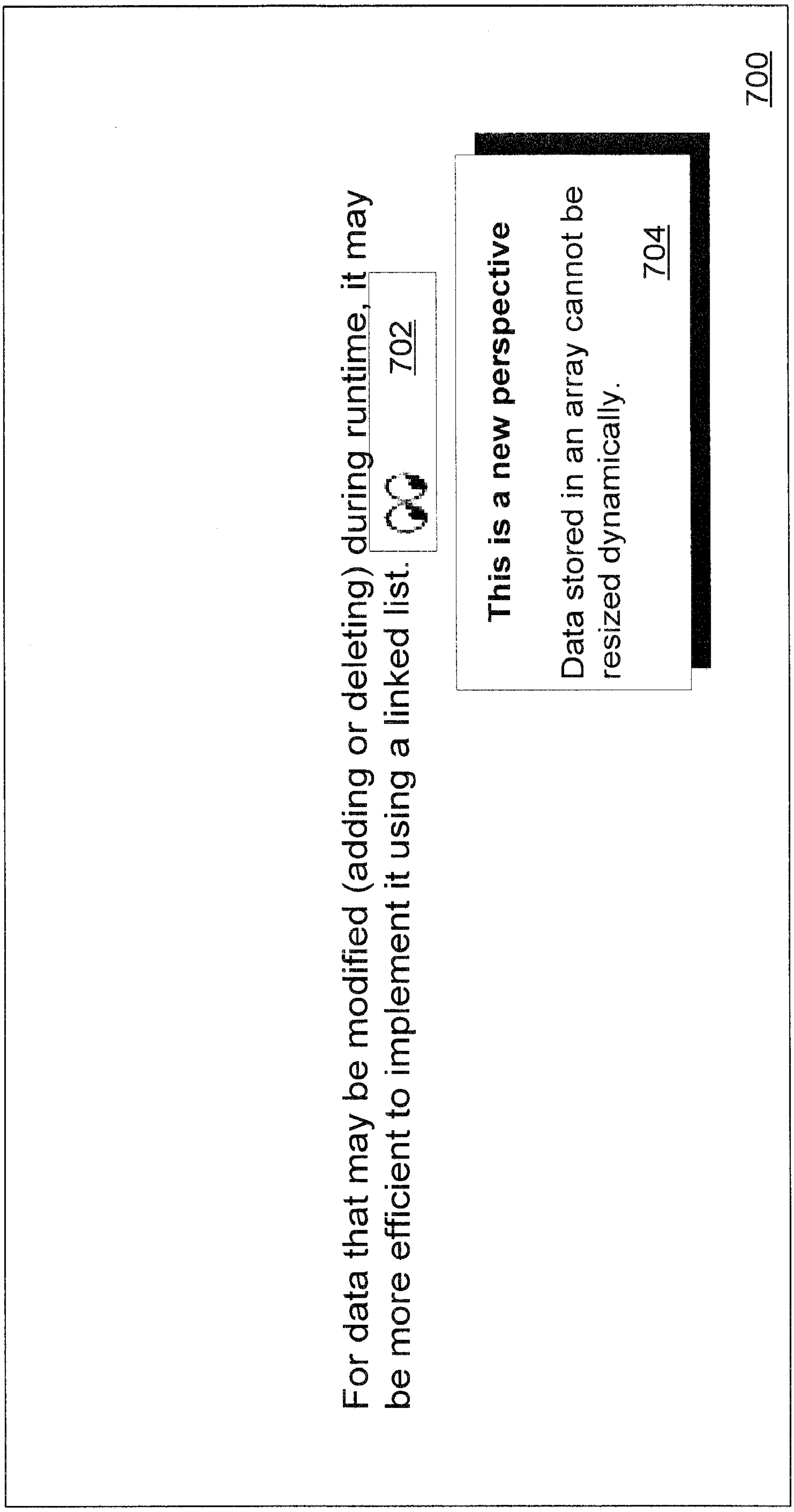
For data that may be modified (adding or deleting) during runtime, it may be more efficient to implement it using a linked list.
702
This is a new perspective
Data stored in an array cannot be resized dynamically.
704
700

FIG. 7

$$f(x) = (3 + 2)x^3$$

$$f(x) = (5)x^3$$

$$f'(x) = 10x^2$$

800

FIG. 8

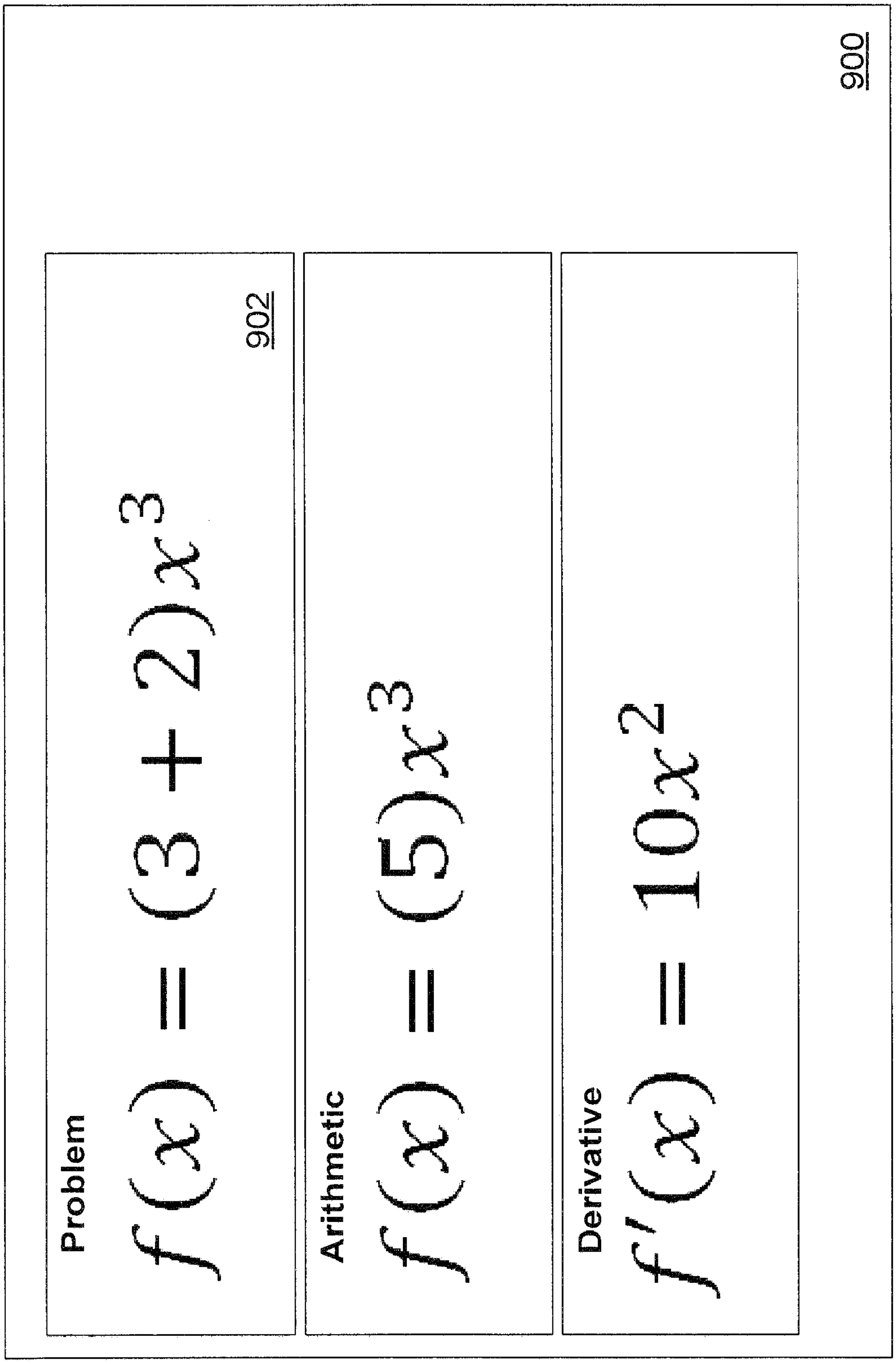
900
Problem
f(x) = (3 + 2)x^3
902
Arithmetic
f(x) = (5)x^3
Derivative
f′(x) = 10x^2

FIG. 9

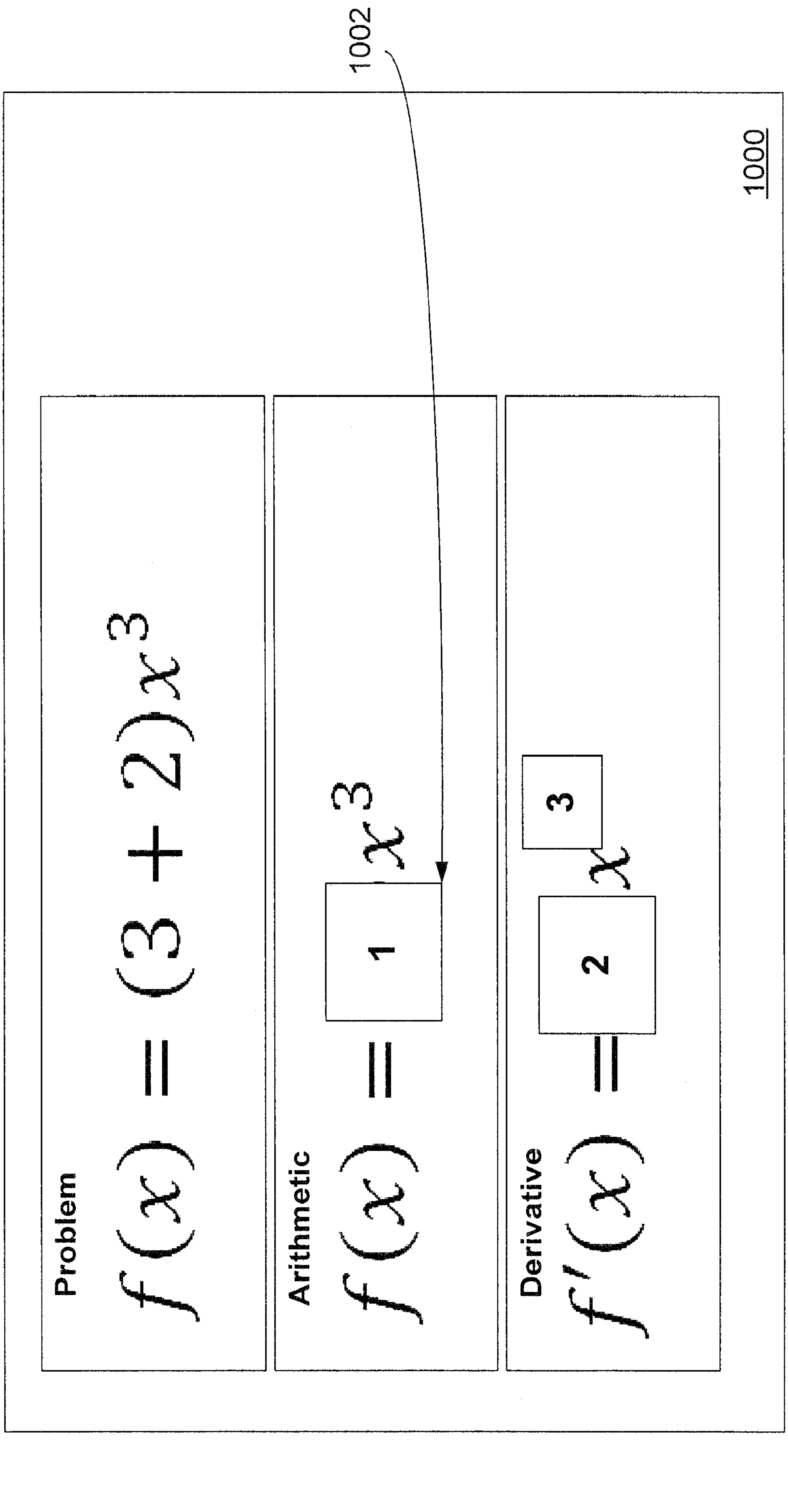
1002
1000
Problem
$f(x) = (3+2)x^3$
Arithmetic
$f(x) = 1\ x^3$
1
Derivative
$f'(x) = 2\ x^{3}$
2
3

FIG. 10

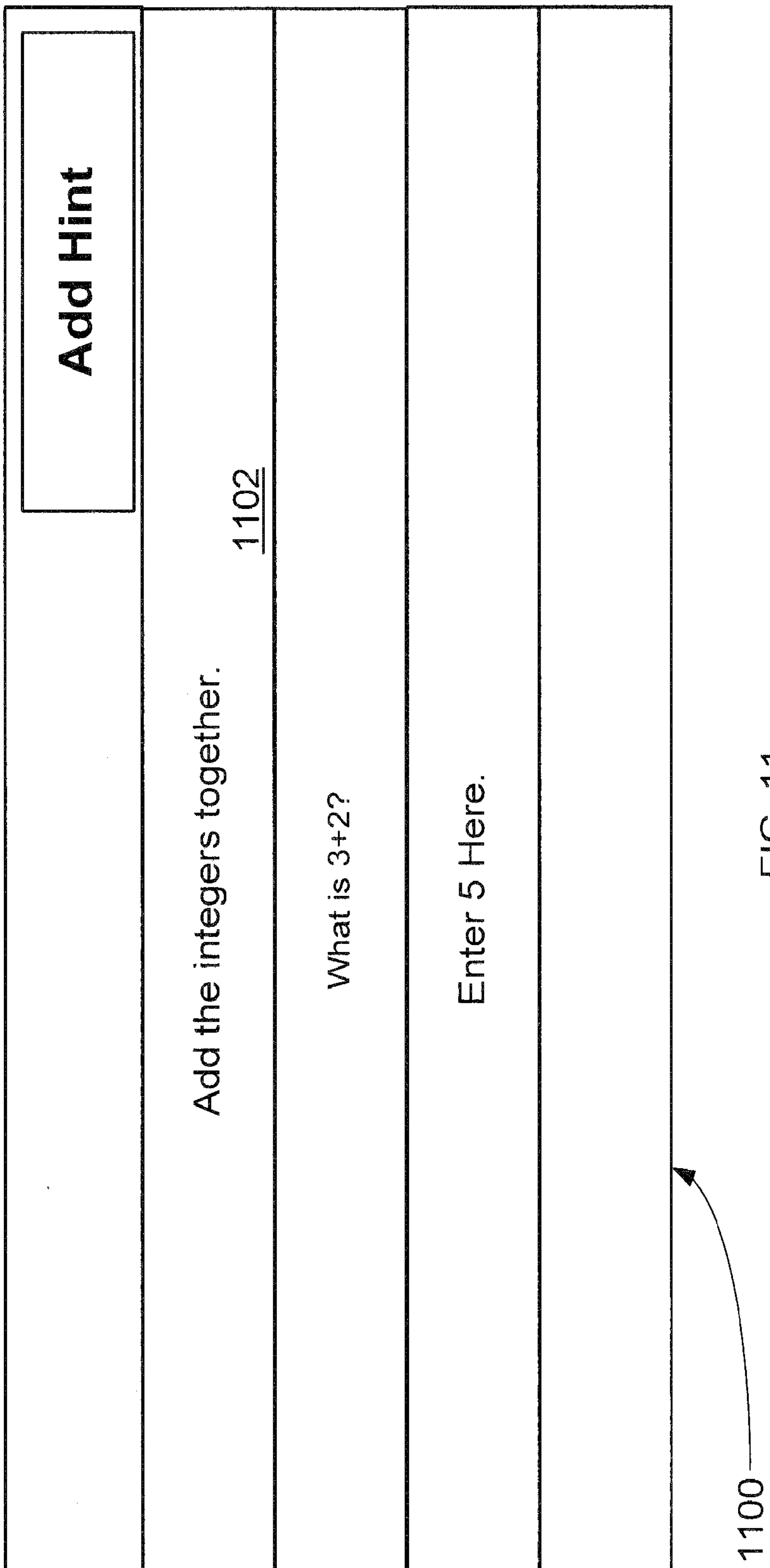
Add Hint
Add the integers together.
1102
What is 3+2?
Enter 5 Here.
1100

FIG. 11

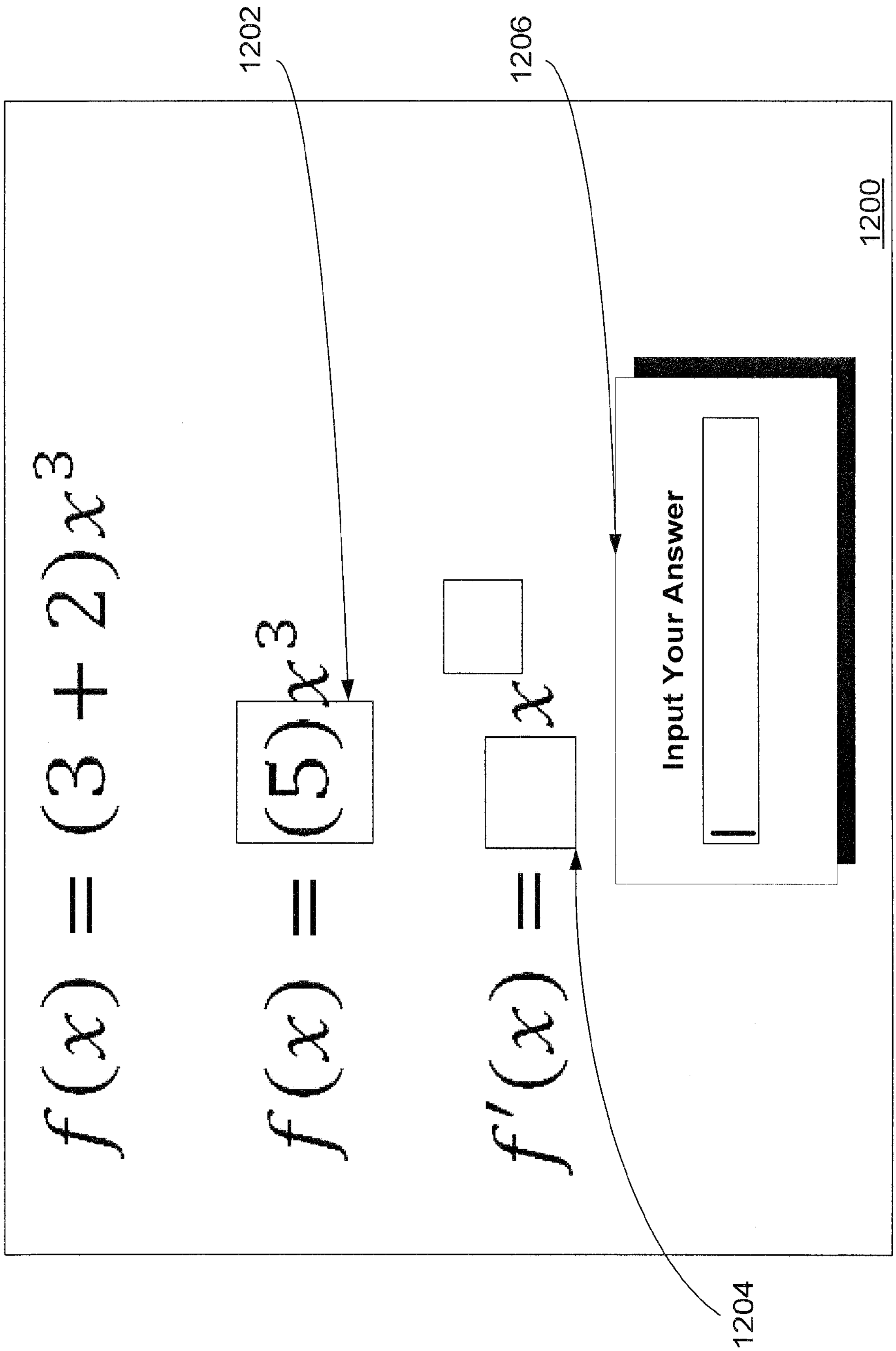
1202
1206
1200
$f(x) = (3+2)x^3$
$f(x) = (5)x^3$
$f'(x) = \square x^{\square}$
Input Your Answer
1204

FIG. 12

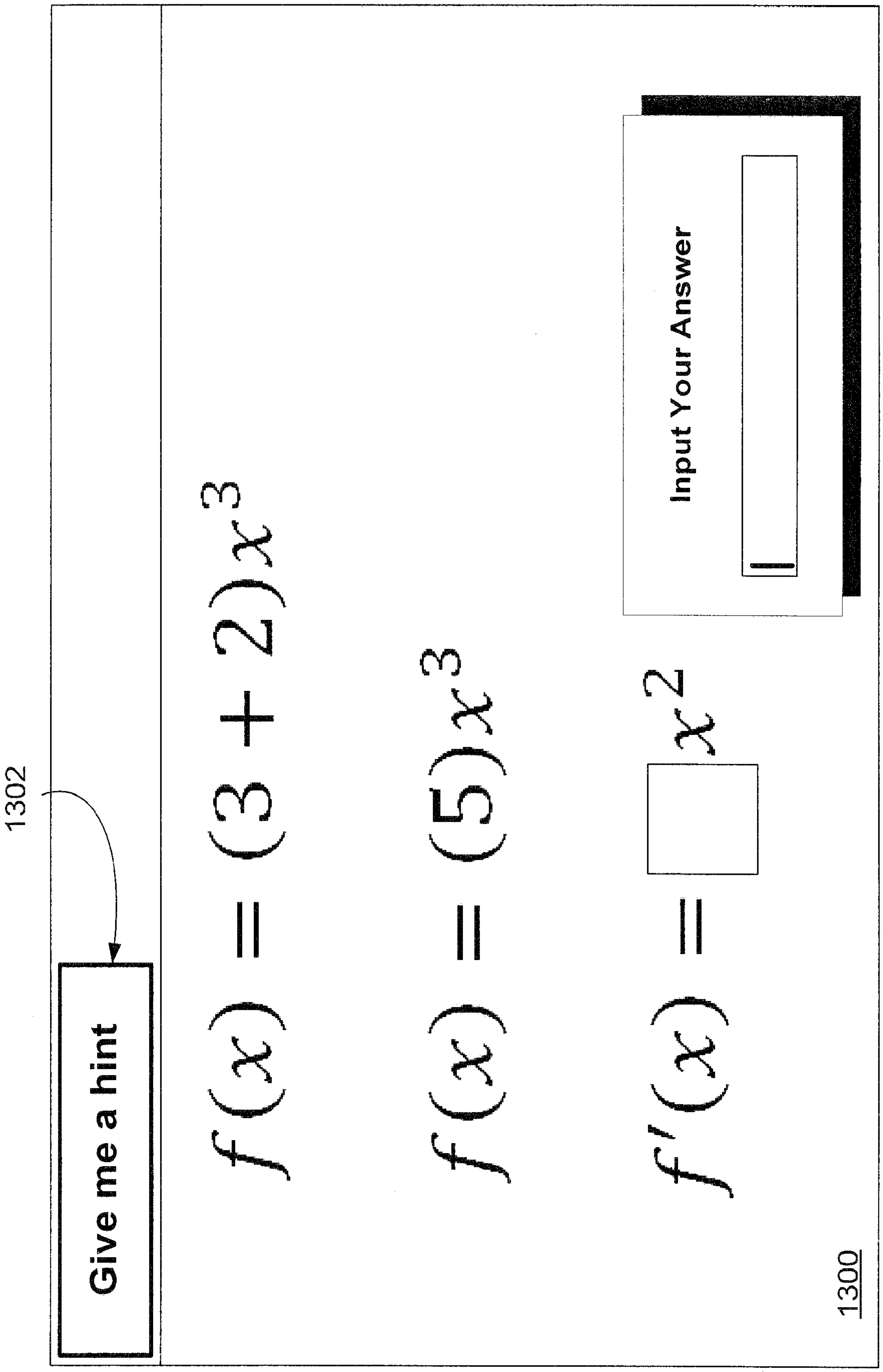

FIG. 13

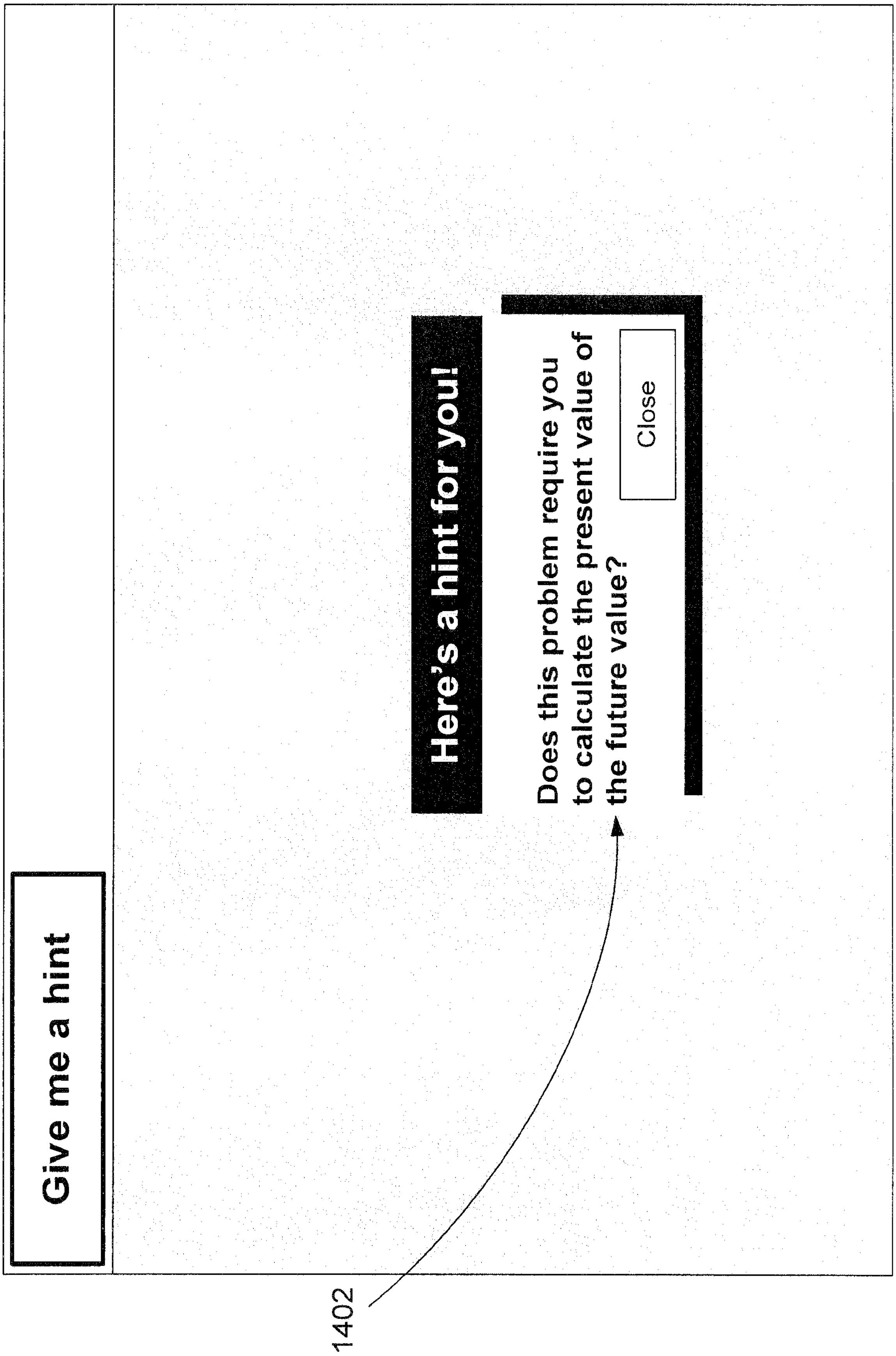
Give me a hint
Here's a hint for you!
Does this problem require you to calculate the present value of the future value?
Close
1402

FIG. 14

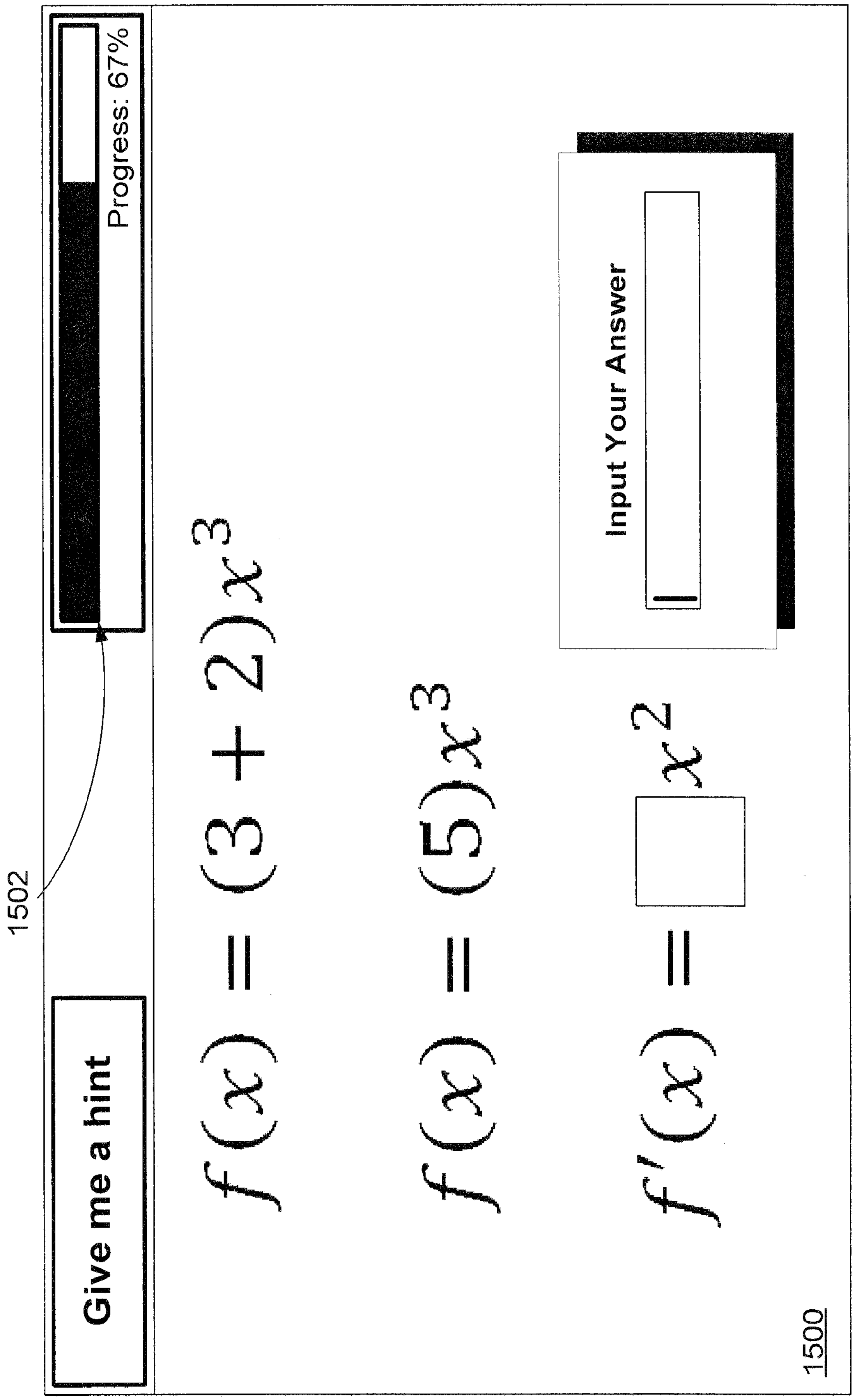
1502
Give me a hint
Progress: 67%
f(x) = (3 + 2)x^3
f(x) = (5)x^3
f′(x) = □x^2
Input Your Answer
1500

FIG. 15

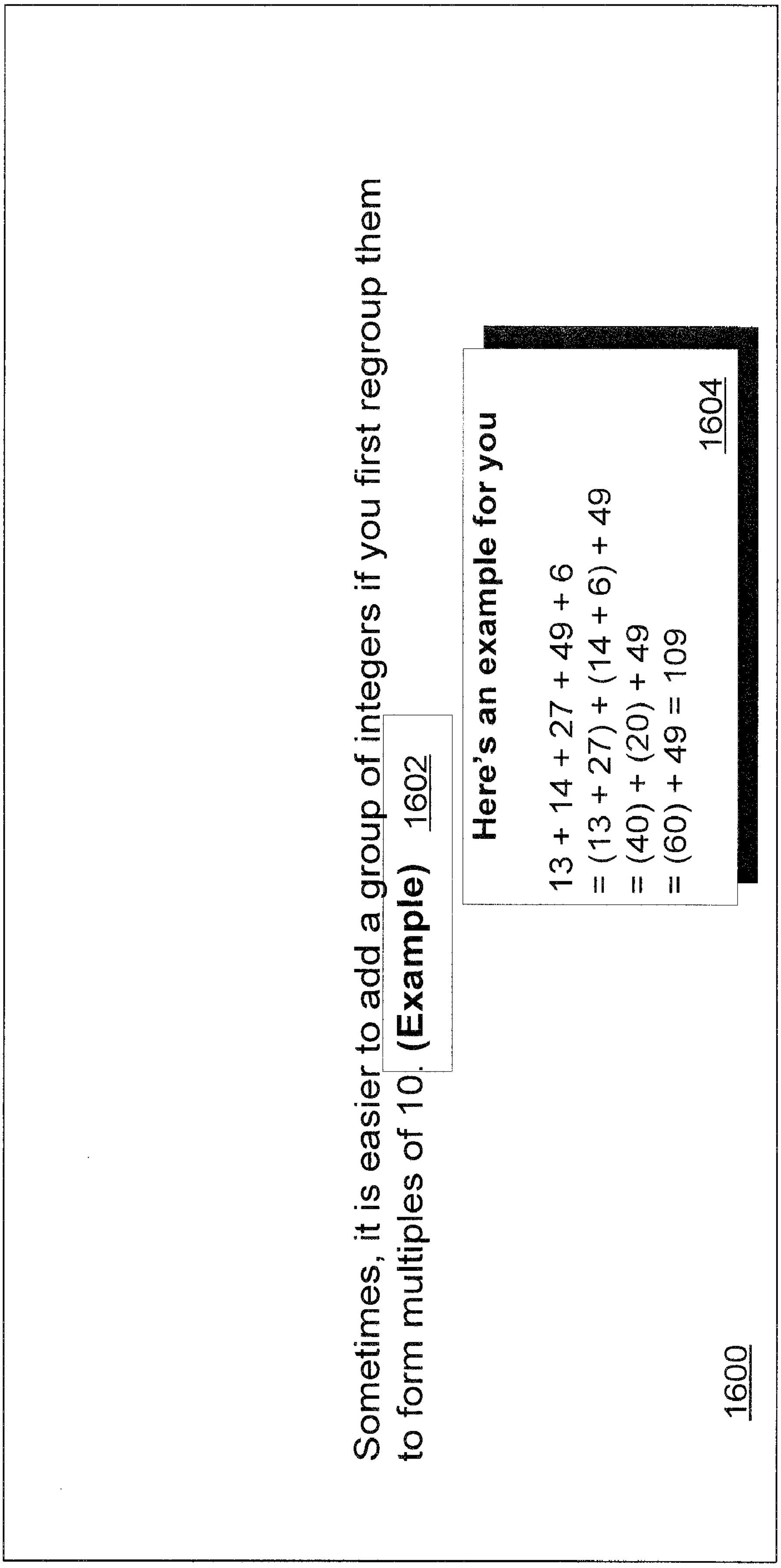
Sometimes, it is easier to add a group of integers if you first regroup them to form multiples of 10.
(Example)
1602
Here's an example for you
13 + 14 + 27 + 49 + 6
= (13 + 27) + (14 + 6) + 49
= (40) + (20) + 49
= (60) + 49 = 109
1604
1600

FIG. 16

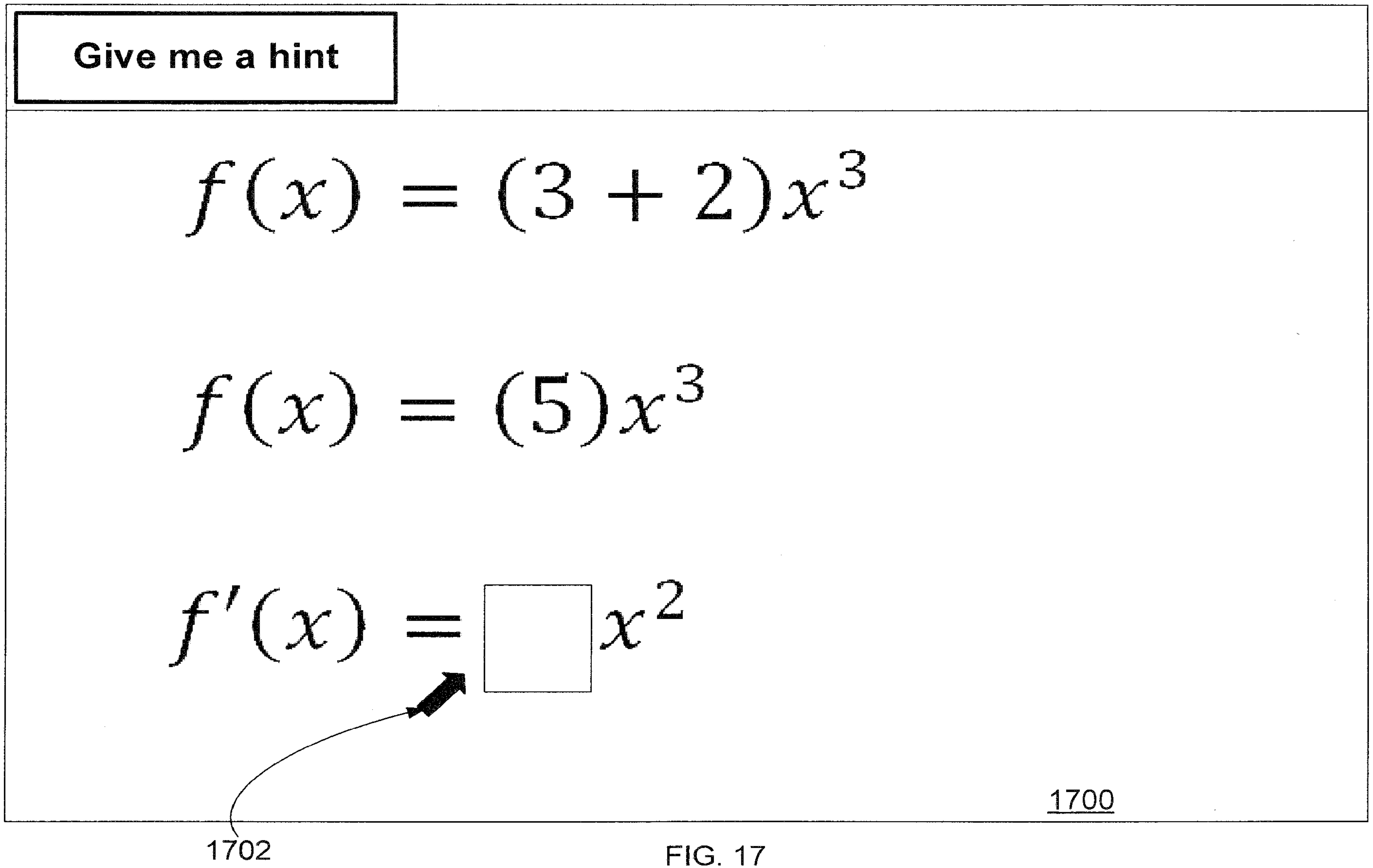

FIG. 17

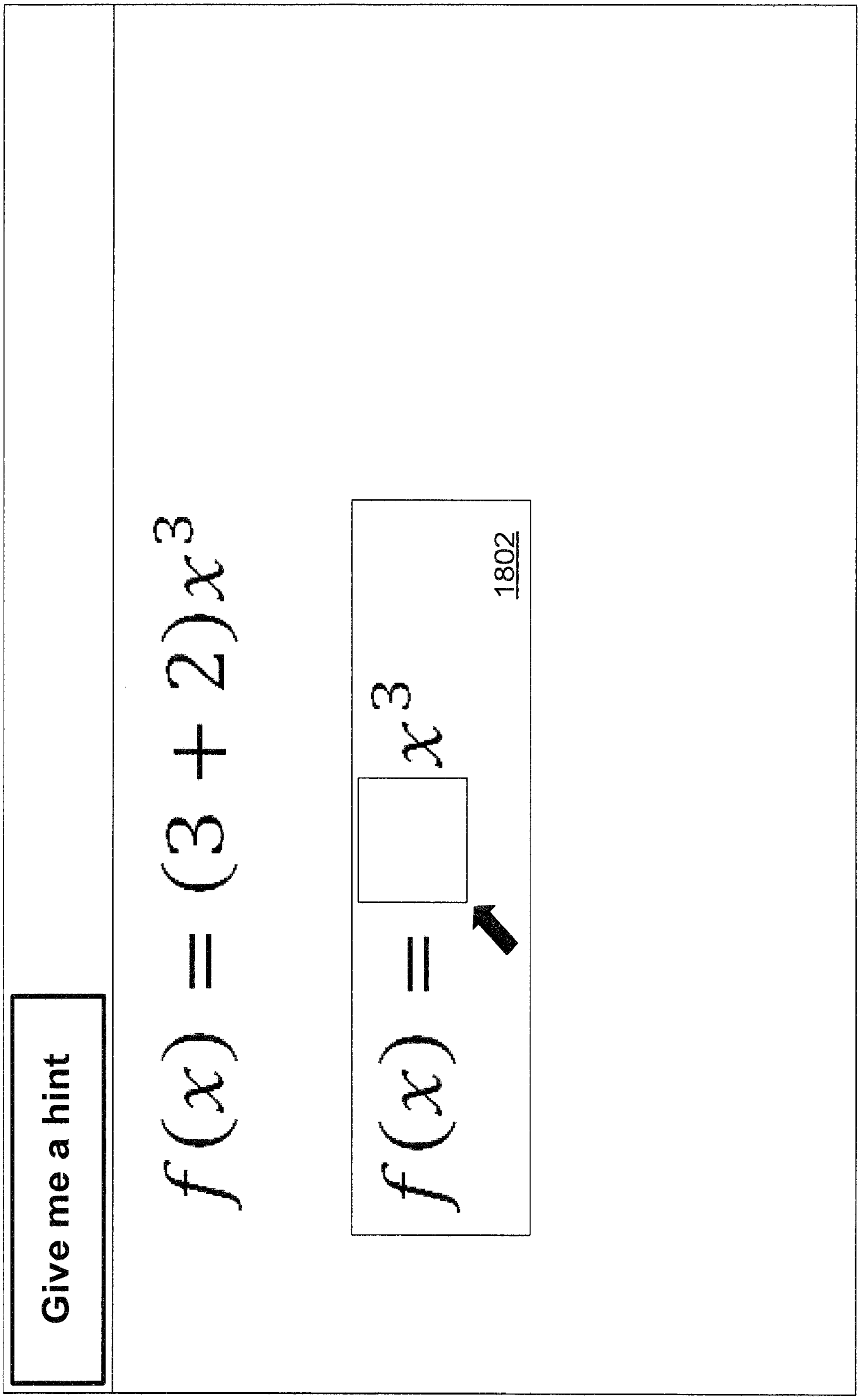

FIG. 18

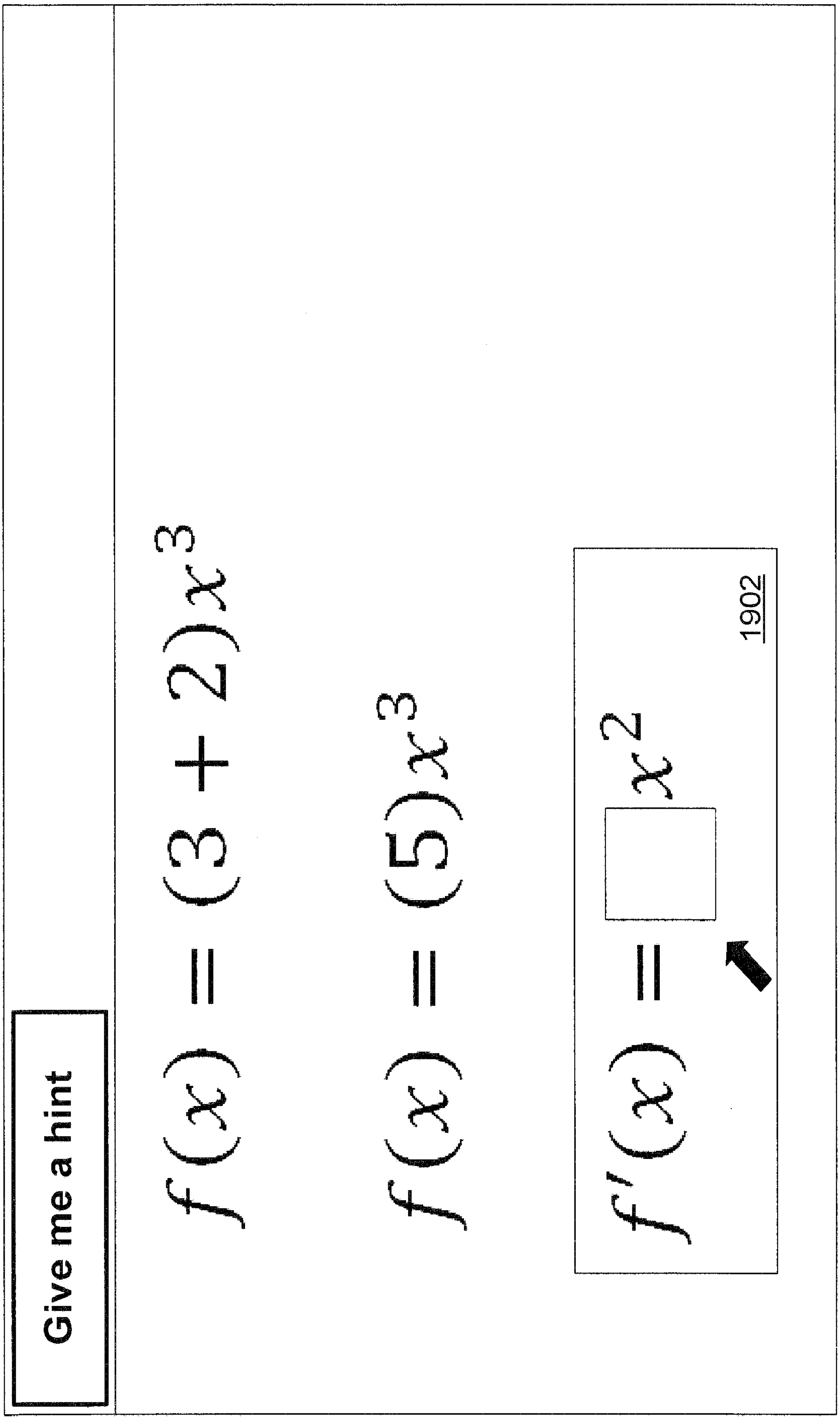

FIG. 19

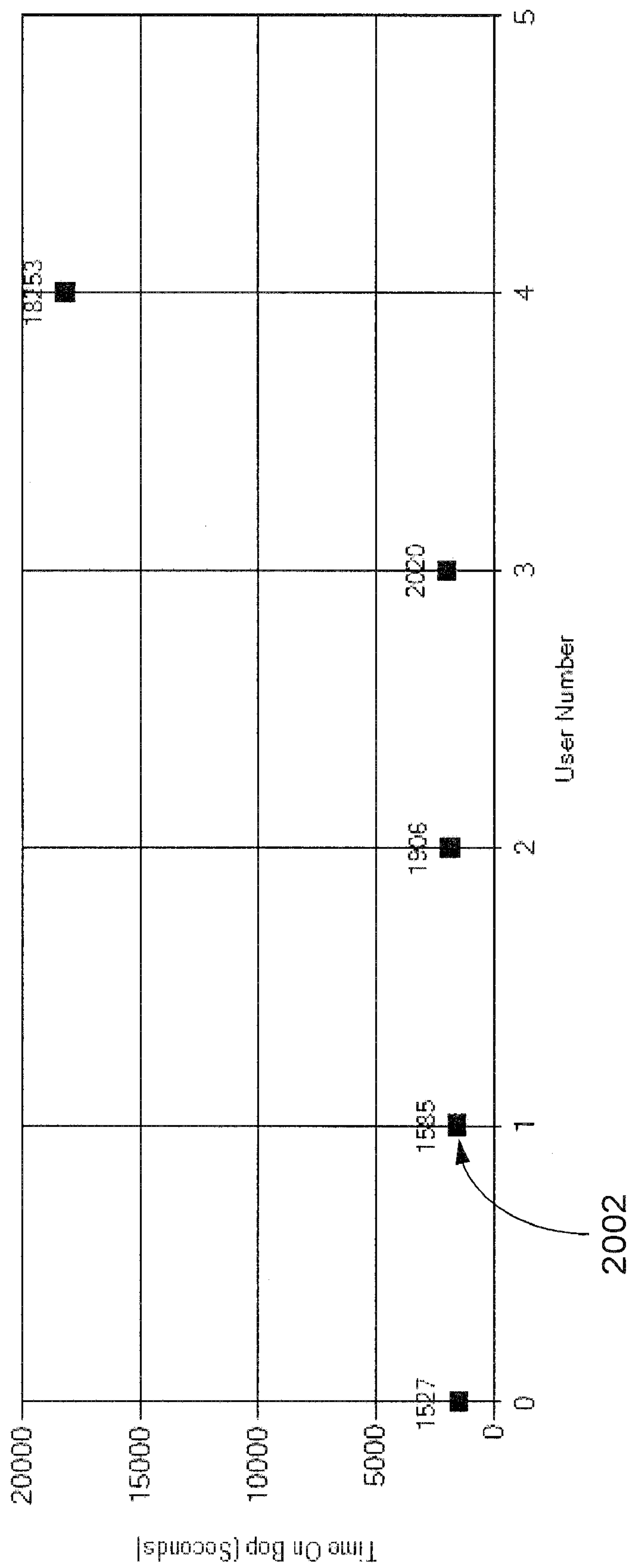
Time On Bop (Seconds)
20000
15000
10000
5000
0
1527
1585
1906
2020
18153
0
1
2
3
4
5
User Number
2002

FIG. 20

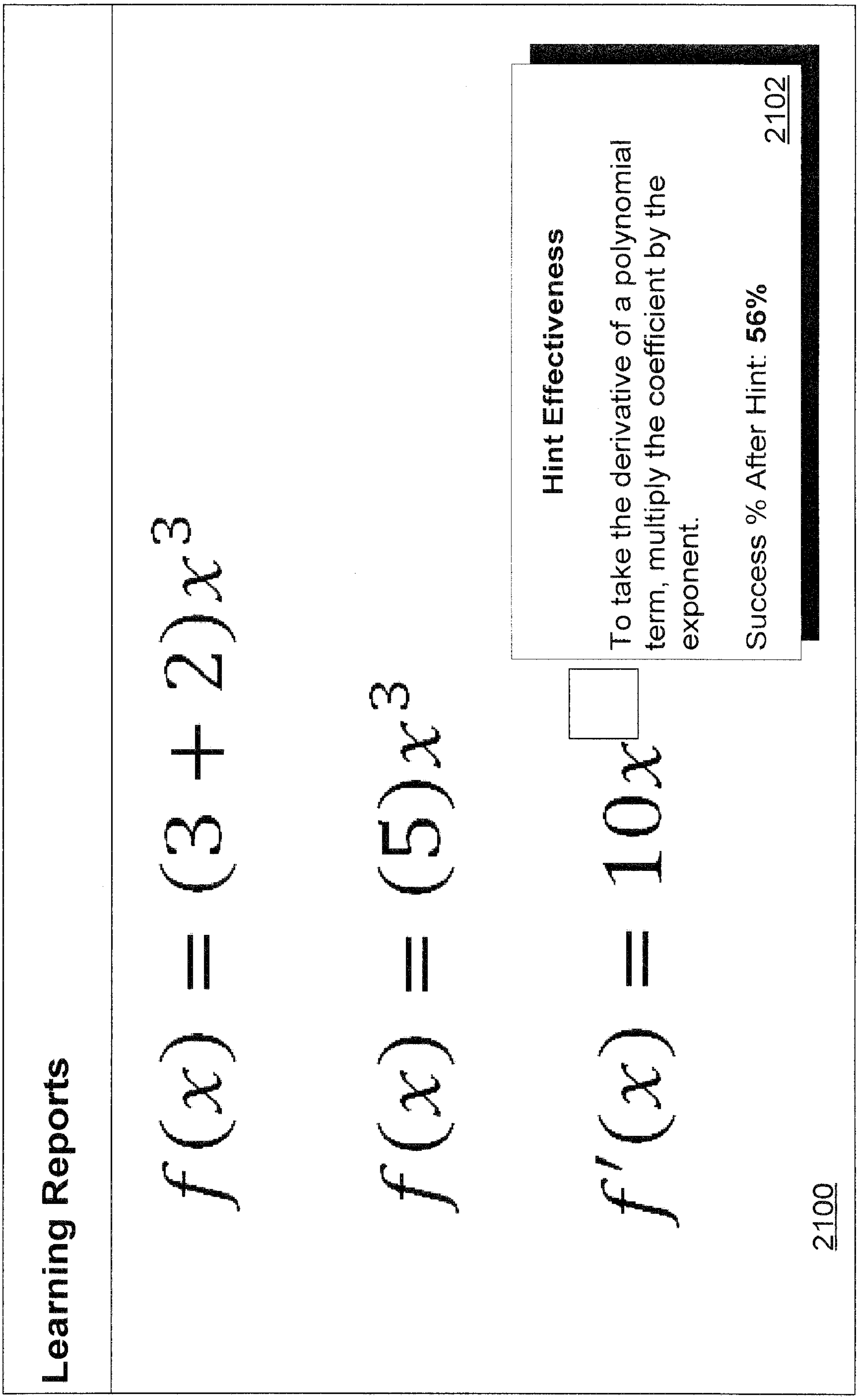
Learning Reports
$f(x) = (3+2)x^3$
$f(x) = (5)x^3$
$f'(x) = 10x^{\square}$
Hint Effectiveness
To take the derivative of a polynomial term, multiply the coefficient by the exponent.
Success % After Hint: 56%
2102
2100

FIG. 21

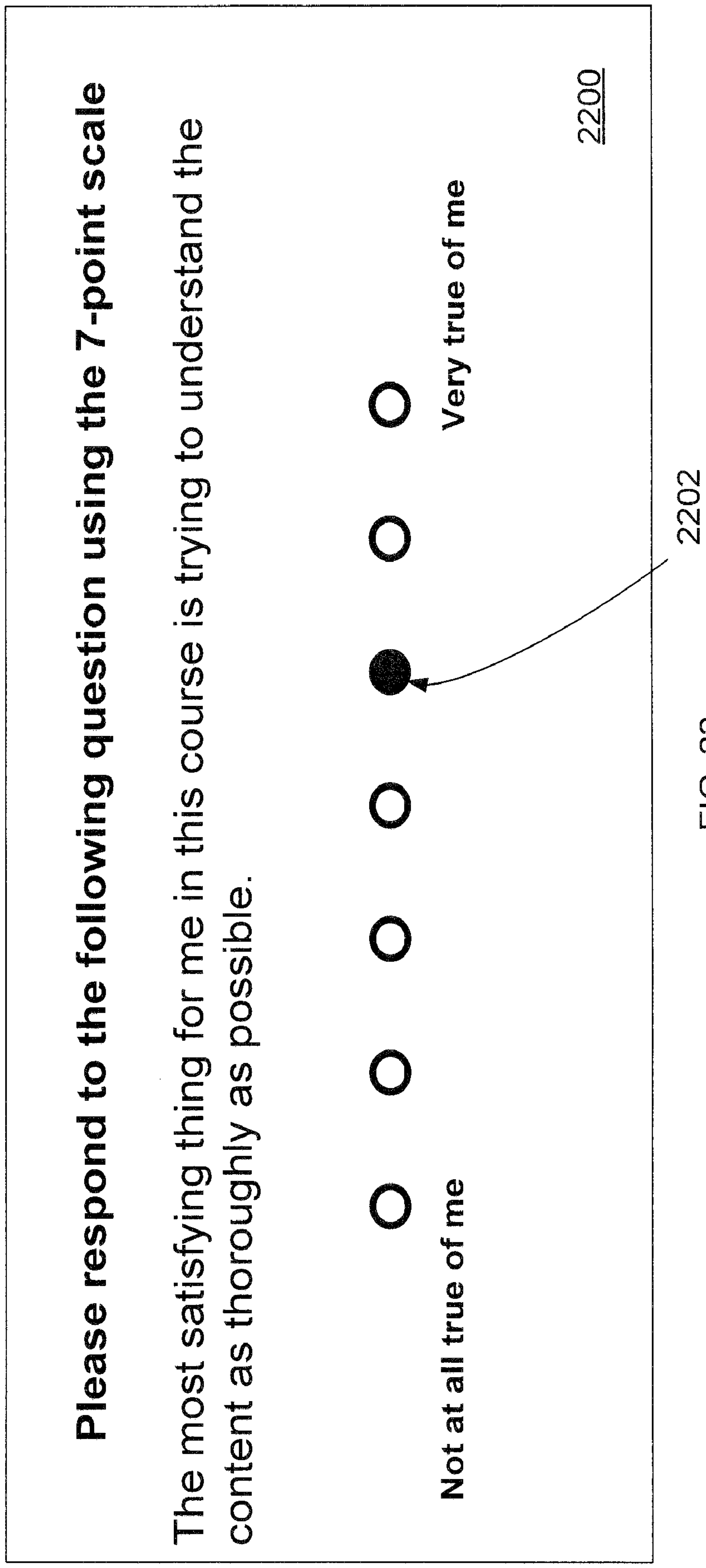
Please respond to the following question using the 7-point scale
The most satisfying thing for me in this course is trying to understand the content as thoroughly as possible.
Not at all true of me
Very true of me
2200
2202

FIG. 22

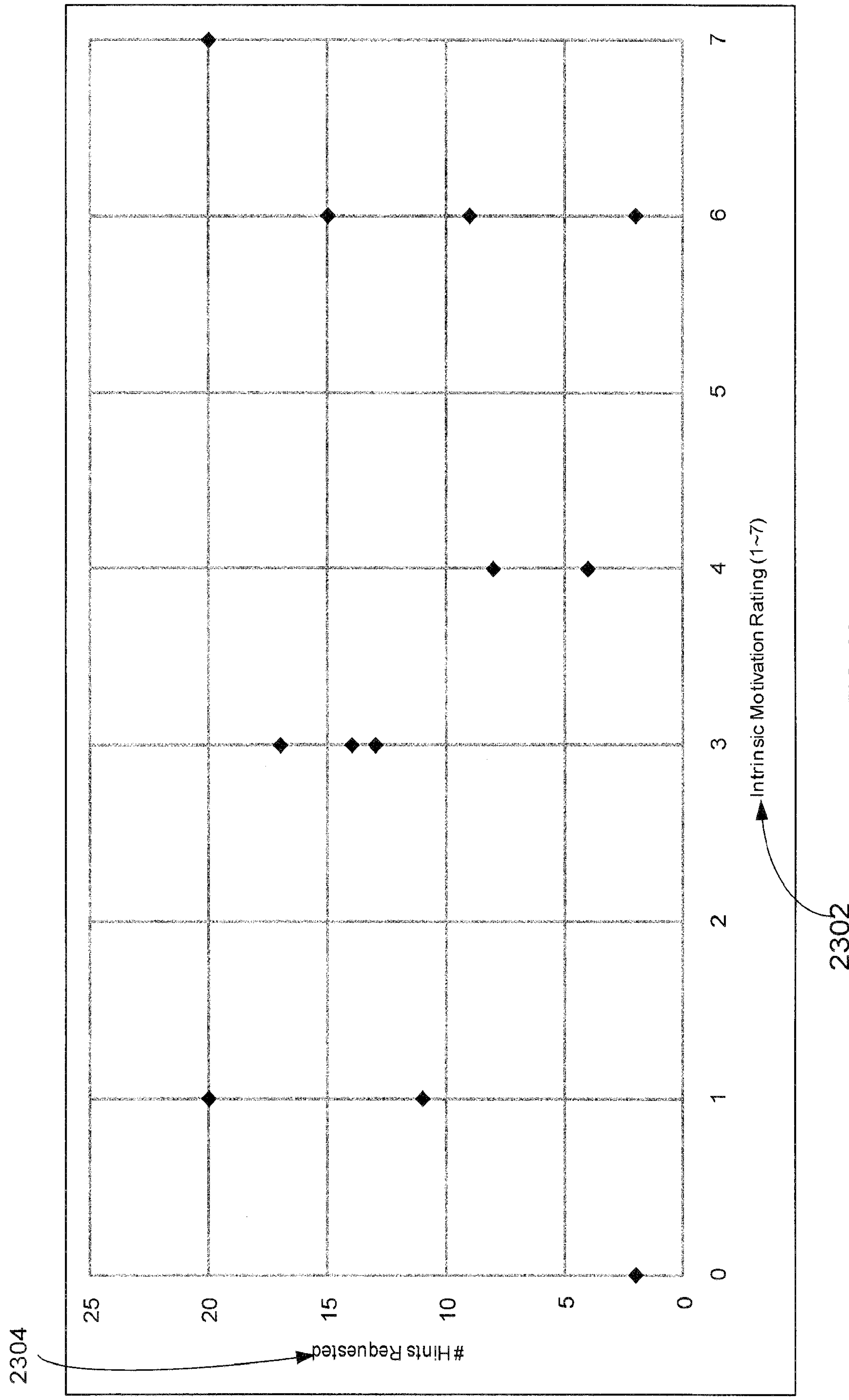
#Hints Requested
25
20
15
10
5
0
Intrinsic Motivation Rating (1~7)
0
1
2
3
4
5
6
7
2302
2304

FIG. 23

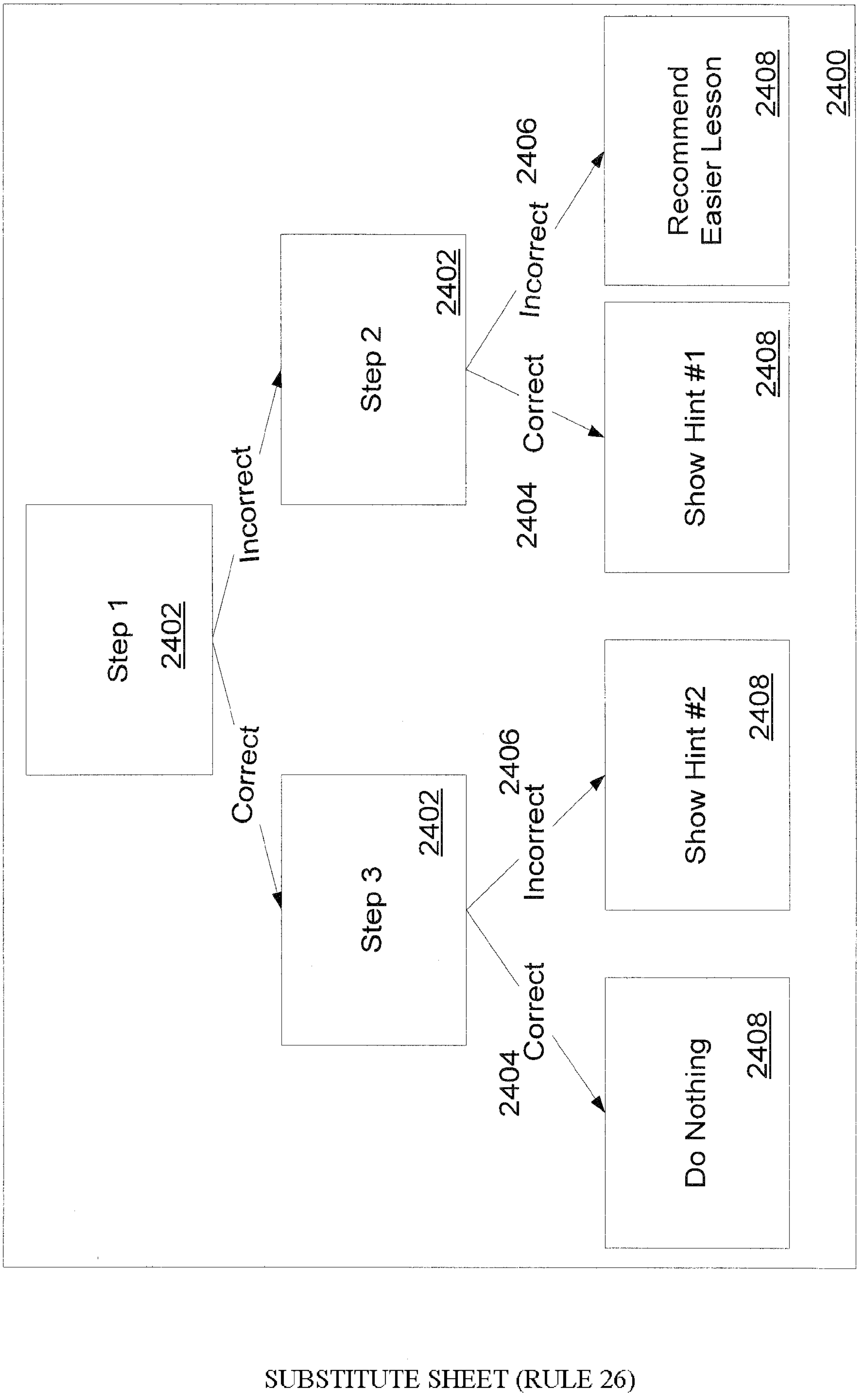
Step 1
2402
Incorrect
Correct
Step 2
2402
Step 3
2402
Incorrect
2406
Correct
2404
Incorrect
2406
Correct
2404
Recommend Easier Lesson
2408
Show Hint #1
2408
Show Hint #2
2408
Do Nothing
2408
2400

FIG. 24

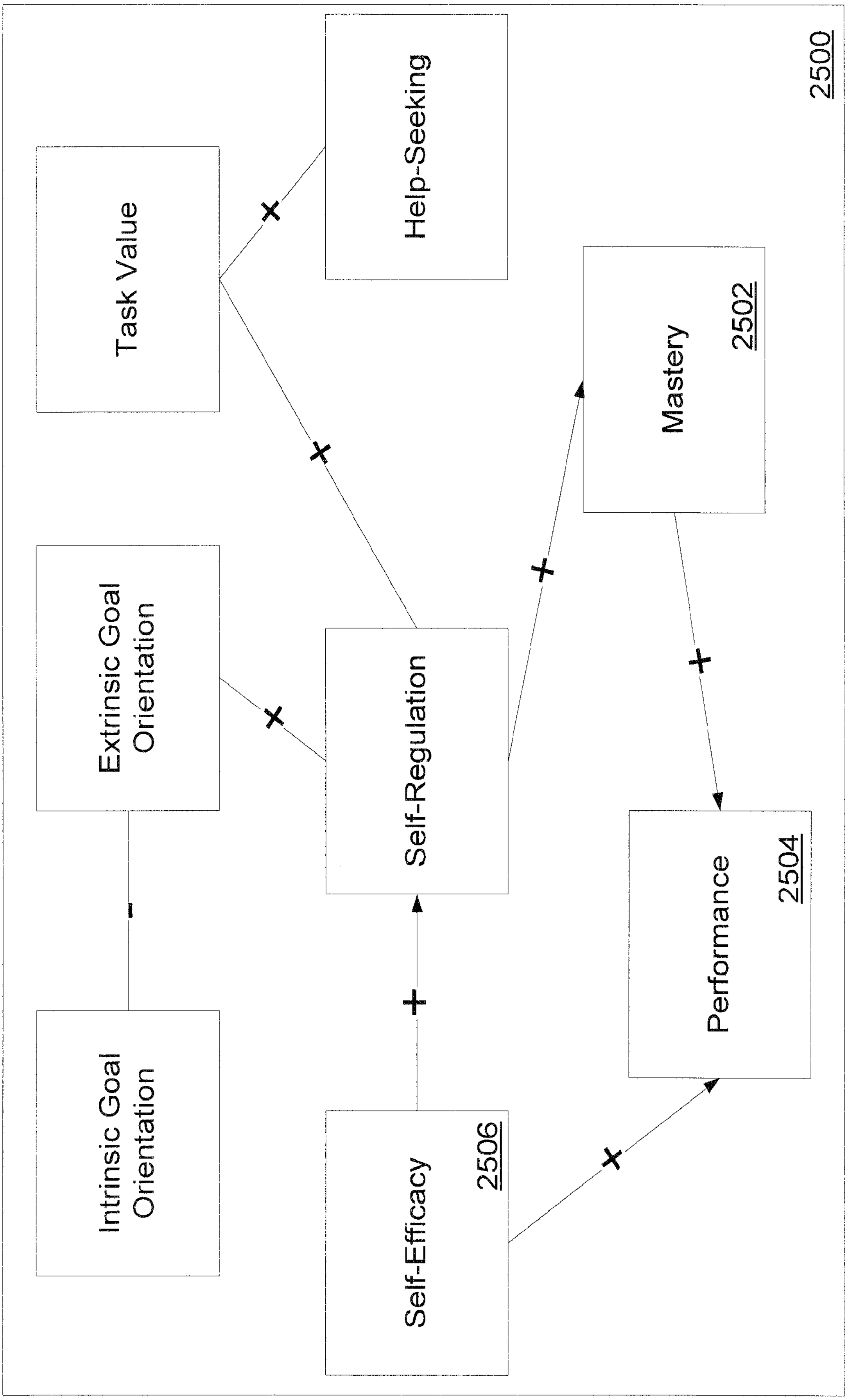
2500
Task Value
Help-Seeking
Extrinsic Goal Orientation
Self-Regulation
Mastery
2502
Intrinsic Goal Orientation
Self-Efficacy
2506
Performance
2504

FIG. 25

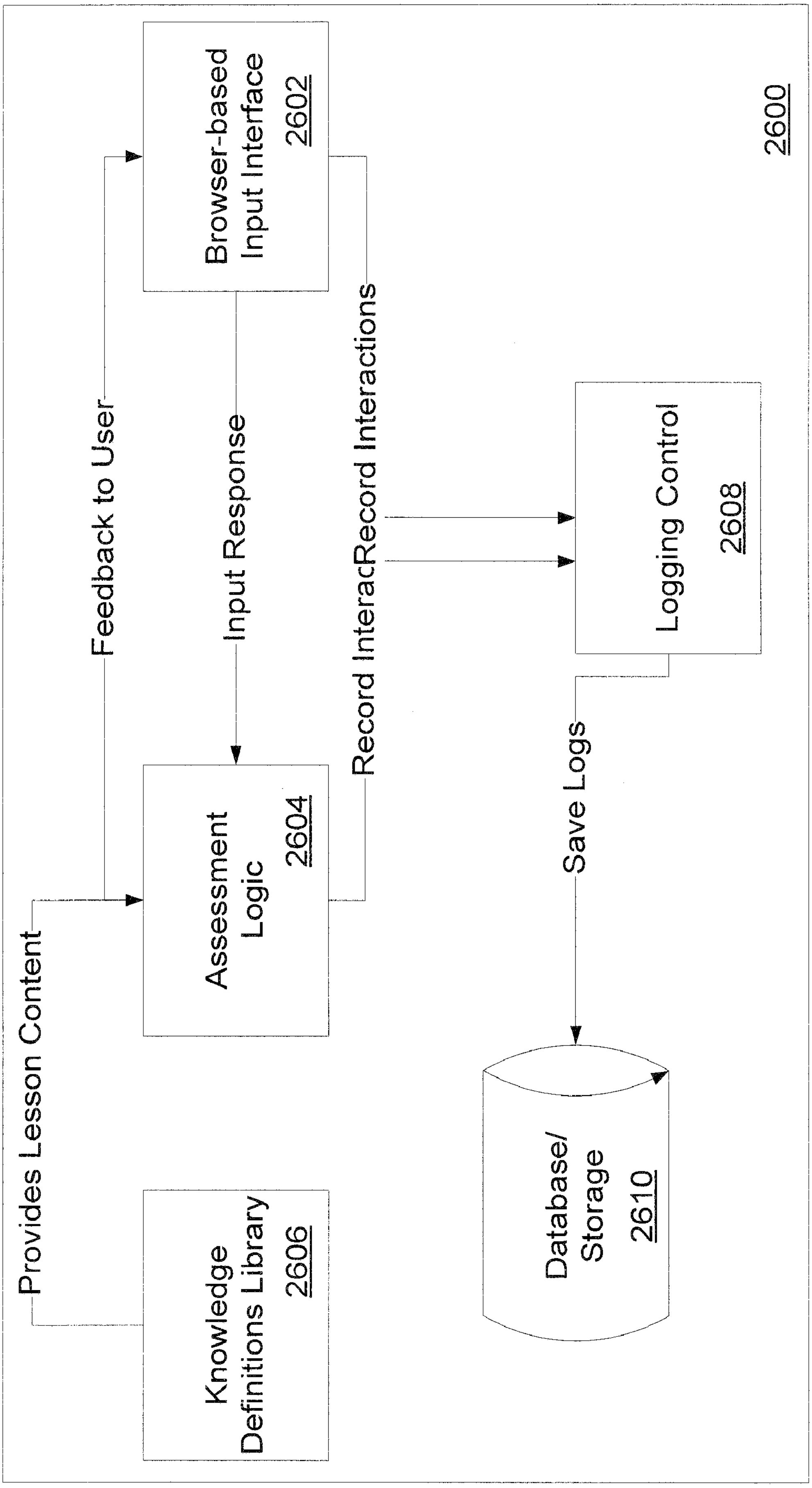
Browser-based Input Interface 2602
2600
Feedback to User
Input Response
Record InteracRecord Interactions
Logging Control 2608
Assessment Logic 2604
Save Logs
Provides Lesson Content
Knowledge Definitions Library 2606
Database/ Storage 2610

FIG. 26

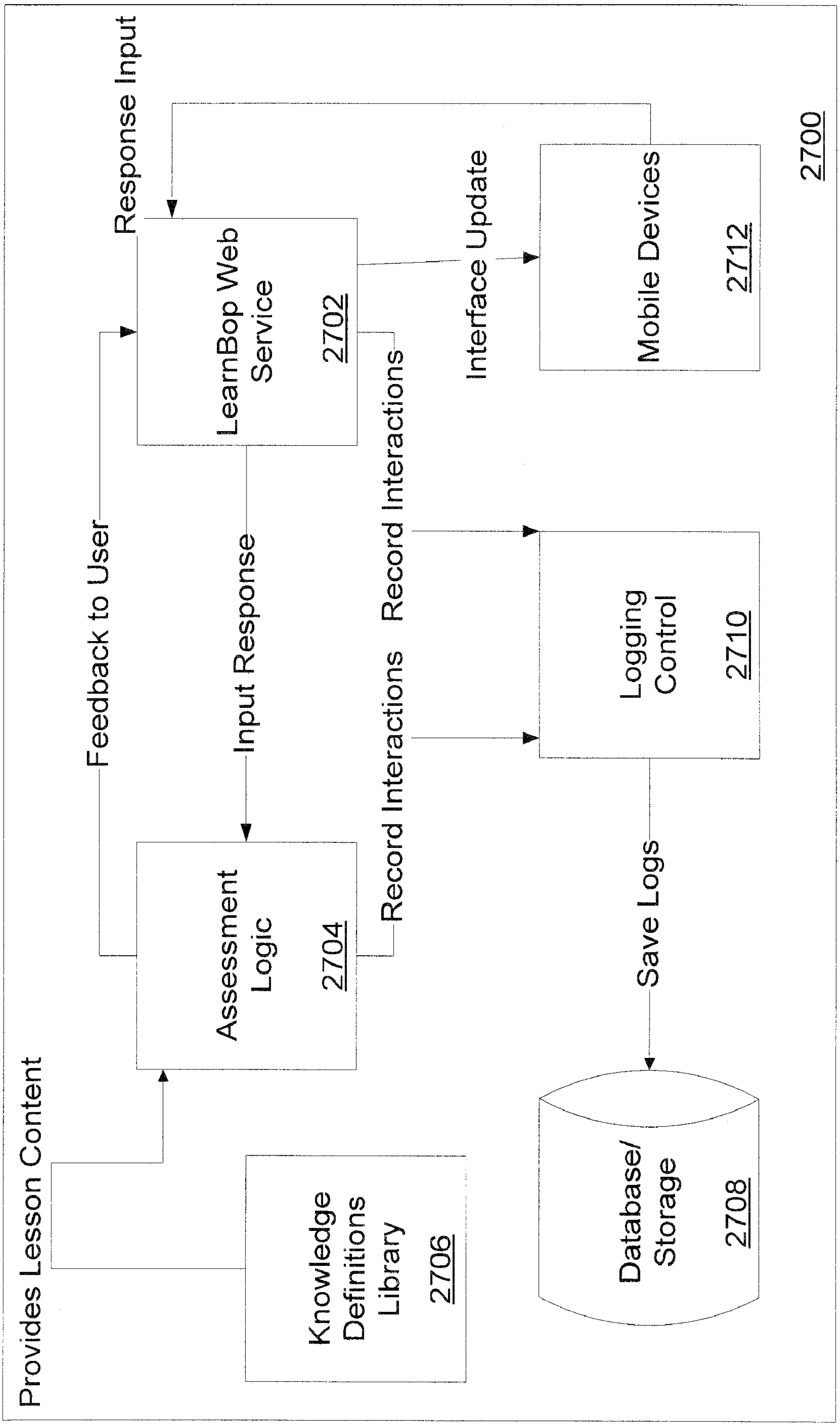
Response Input
LearnBop Web Service
2702
Interface Update
Mobile Devices
2712
2700
Feedback to User
Input Response
Record Interactions
Record Interactions
Logging Control
2710
Assessment Logic
2704
Save Logs
Provides Lesson Content
Knowledge Definitions Library
2706
Database/ Storage
2708

FIG. 27

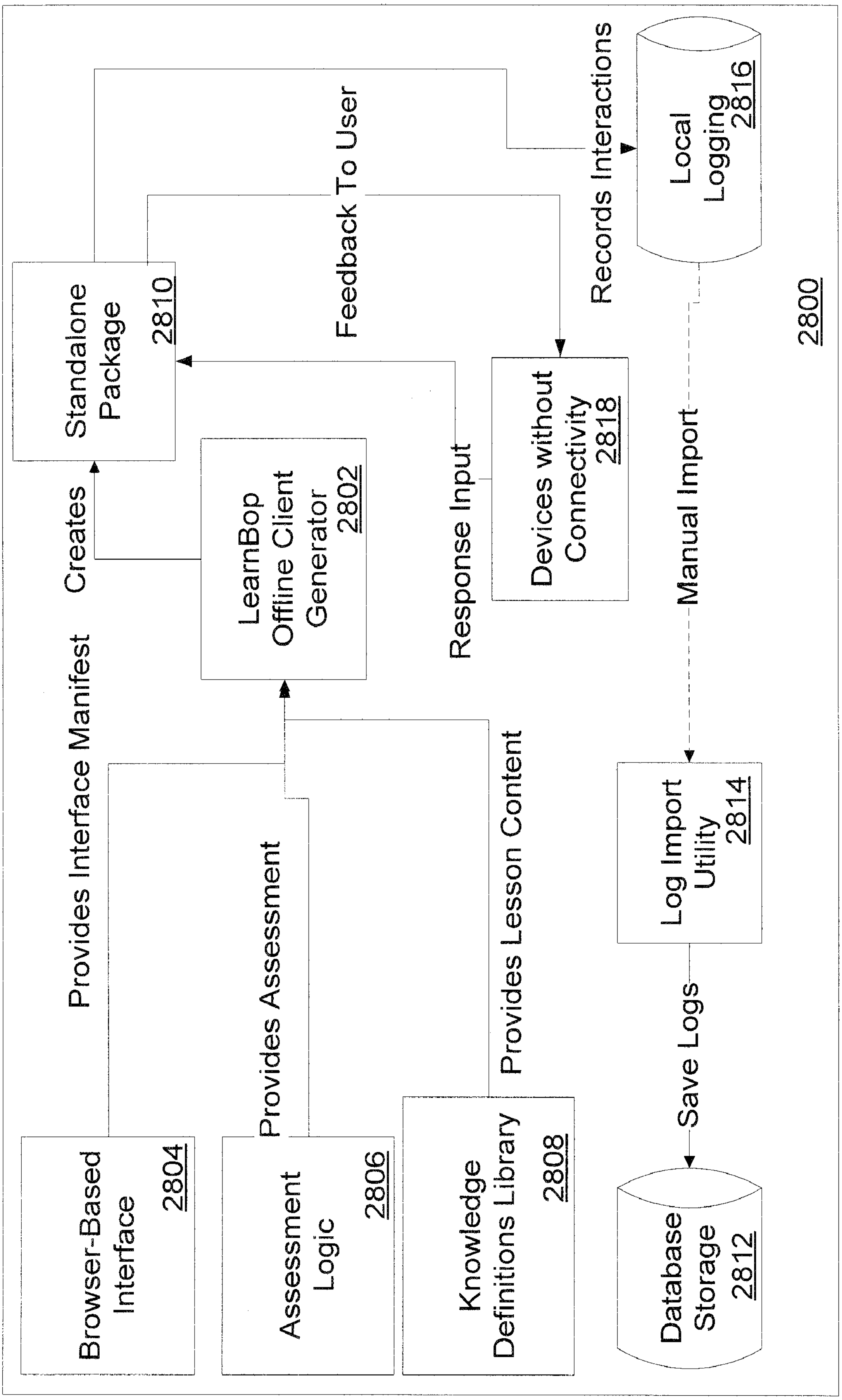
Browser-Based Interface 2804
Assessment Logic 2806
Knowledge Definitions Library 2808
Database Storage 2812
Provides Interface Manifest
Provides Assessment
Provides Lesson Content
Save Logs
LearnBop Offline Client Generator 2802
Log Import Utility 2814
Creates
Response Input
Manual Import
Standalone Package 2810
Devices without Connectivity 2818
Feedback To User
2800
Records Interactions
Local Logging 2816

FIG. 28

INTERNATIONAL SEARCH REPORT PCT/US2012/029551 13.06.2012

International application No.
PCT/US 12/29551

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G09B 3/00 (2012.01)
USPC - 434/350
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
USPC: 434/350

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 434/350, 365; 700/1, 28, 47, 89, 90 (keyword limited - see search terms below)

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PubWEST (PGPB, USPT, USOC, EPAB, JPAB); Google Web; Google Scholar
Terms: learning, interactive, student, education, teach, content, media, feedback, adaptive, progress, report, reusable, hypertext, author.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2011/0029591 A1 (Wood et al.) 03 February 2011 (03.02.2011), entire document, especially abstract, para [0014], [0032], [0033], [0041], [0043], [0047], [0053], [0054], [0070], [0071], [0091], [0109], [0124].	1-17
Y	US 2008/0160491 A1 (Allen et al.) 03 July 2008 (03.07.2008), entire document, especially abstract, para [0006], [0011], [0056], [0069], [0072].	1-17
A	US 2010/0190143 A1 (Gal et al.) 29 July 2010 (29.07.2010), entire document, especially abstract, para [0055], [0056], [0064], [0065], [0074], [0081].	1-17

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
02 June 2012 (02.06.2012)	13 JUN 2012

Name and mailing address of the ISA/US	Authorized officer:
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774