United States Patent Application Publication	20250252264
Kind Code	A1
Publication Date	August 07, 2025
Inventor(s)	Paley; Andrew R. et al.

Applied Artificial Intelligence Technology for Narrative Generation Based on a Conditional Outcome Framework

Abstract

Artificial intelligence (AI) technology can be used in combination with composable communication goal statements to facilitate a user's ability to quickly structure story outlines in a manner usable by an NLG narrative generation system without any need for the user to directly author computer code. Narrative analytics that are linked to communication goal statements can employ a conditional outcome framework that allows the content and structure of resulting narratives to intelligently adapt as a function of the nature of the data under consideration. This AI technology permits NLG systems to determine the appropriate content for inclusion in a narrative story about a data set in a manner that will satisfy a desired communication goal.

Inventors: **Paley; Andrew R. (Chicago, IL), Nichols; Nathan D. (Chicago, IL), Trahan; Matthew L. (Chicago, IL), Lewis Meza; Maia (Chicago, IL), Pham; Michael Tien Thinh (Chicago, IL), Truong; Charlie M. (Aurora, IL)**

Applicant: **Salesforce, Inc.** (San Francisco, CA)

Family ID: **74851739**

Appl. No.: **19/191626**

Filed: **April 28, 2025**

Related U.S. Application Data

parent US continuation 18616731 20240326 parent-grant-document US 12314674 child US 19191626
parent US continuation 18152304 20230110 parent-grant-document US 11983503 child US 18616731
parent US continuation 17191362 20210303 parent-grant-document US 11562146 child US 18152304
parent US continuation 16047837 20180727 parent-grant-document US 10943069 child US 17191362

parent US continuation-in-part 15897331 20180215 parent-grant-document US 10762304 child US 16047837
parent US continuation-in-part 15897350 20180215 parent-grant-document US 10585983 child US 16047837
parent US continuation-in-part 15897359 20180215 parent-grant-document US 10755053 child US 16047837
parent US continuation-in-part 15897373 20180215 parent-grant-document US 10719542 child US 16047837
parent US continuation-in-part 15897364 20180215 parent-grant-document US 10572606 child US 16047837
parent US continuation-in-part 15897381 20180215 parent-grant-document US 10713442 child US 16047837
us-provisional-application US 62539832 20170801
us-provisional-application US 62460349 20170217
us-provisional-application US 62460349 20170217
us-provisional-application US 62460349 20170217
us-provisional-application US 62460349 20170217
us-provisional-application US 62460349 20170217
us-provisional-application US 62460349 20170217

Publication Classification

Int. Cl.: **G06F40/30** (20200101); **G06F40/295** (20200101); **G06N5/02** (20230101); **G10L17/18** (20130101)

U.S. Cl.:

CPC **G06F40/30** (20200101); **G06F40/295** (20200101); **G06N5/02** (20130101); **G10L17/18** (20130101);

Background/Summary

CROSS-REFERENCE TO RELATED PATENT APPLICATIONS [0001] This patent application is a continuation of U.S. patent application Ser. No. 18/616,731 (Attorney Docket No. SFDCP845USC2), filed Mar. 26, 2024, titled “Applied Artificial Intelligence Technology for Narrative Generation Based on a Conditional Outcome Framework”, which is a continuation of U.S. patent application Ser. No. 18/152,304 (Attorney Docket No. SFDCP845C1), filed Jan. 10, 2023, titled “Applied Artificial Intelligence Technology for Narrative Generation Based on a Conditional Outcome Framework”, which is a continuation of U.S. patent application Ser. No. 17/191,362, filed Mar. 3, 2021, titled “Applied Artificial Intelligence Technology for Narrative Generation Based on a Conditional Outcome Framework”, now U.S. Pat. No. 11,562,146, which is a continuation of U.S. patent application Ser. No. 16/047,837, filed Jul. 27, 2018, titled “Applied Artificial Intelligence Technology for Narrative Generation Based on a Conditional Outcome Framework”, now U.S. Pat. No. 10,943,069, which (1) claims priority to U.S. provisional patent application Ser. No. 62/539,832, filed Aug. 1, 2017, and titled “Applied Artificial Intelligence Technology for Narrative Generation Based on Analysis Communication Goals”, and (2) is also a continuation-in-part of (i) U.S. patent application Ser. No. 15/897,331, filed Feb. 15, 2018, titled “Applied Artificial Intelligence Technology for Performing Natural Language Generation (NLG) Using Composable Communication Goals and Ontologies to Generate Narrative Stories”, now U.S.

Pat. No. 10,762,304, (ii) U.S. patent application Ser. No. 15/897,350, filed Feb. 15, 2018, titled "Applied Artificial Intelligence Technology for Determining and Mapping Data Requirements for Narrative Stories to Support Natural Language Generation (NLG) Using Composable Communication Goals", now U.S. Pat. No. 10,585,983, (iii) U.S. patent application Ser. No. 15/897,359, filed Feb. 15, 2018, titled "Applied Artificial Intelligence Technology for Story Outline Formation Using Composable Communication Goals to Support Natural Language Generation (NLG)", now U.S. Pat. No. 10,755,053, (iv) U.S. patent application Ser. No. 15/897,364, filed Feb. 15, 2018, titled "Applied Artificial Intelligence Technology for Runtime Computation of Story Outlines to Support Natural Language Generation (NLG)", now U.S. Pat. No. 10,572,606, (v) U.S. patent application Ser. No. 15/897,373, filed Feb. 15, 2018, titled "Applied Artificial Intelligence Technology for Ontology Building to Support Natural Language Generation (NLG) Using Composable Communication Goals", now U.S. Pat. No. 10,719,542, and (vi) U.S. patent application Ser. No. 15/897,381, filed Feb. 15, 2018, titled "Applied Artificial Intelligence Technology for Interactive Story Editing to Support Natural Language Generation (NLG)", now U.S. Pat. No. 10,713,442, each of which claims priority to U.S. provisional patent application Ser. No. 62/460,349, filed Feb. 17, 2017, and titled "Applied Artificial Intelligence Technology for Performing Natural Language Generation (NLG) Using Composable Communication Goals and Ontologies to Generate Narrative Stories", the entire disclosures of each of which are incorporated herein by reference in their entirety and for all purposes. [0002] This patent application is related to U.S. patent application Ser. No. 16/047,800, filed Jul. 27, 2018, titled "Applied Artificial Intelligence Technology for Narrative Generation Based on Analysis Communication Goals", now U.S. Pat. No. 10,699,079, the entire disclosure of which is incorporated herein by reference in its entirety and for all purposes.

INTRODUCTION

[0003] There is an ever-growing need in the art for improved natural language generation (NLG) technology that harnesses computers to process data sets and automatically generate narrative stories about those data sets. NLG is a subfield of artificial intelligence (AI) concerned with technology that produces language as output on the basis of some input information or structure, in the cases of most interest here, where that input constitutes data about some situation to be analyzed and expressed in natural language. Many NLG systems are known in the art that use template approaches to translate data into text. However, such conventional designs typically suffer from a variety of shortcomings such as constraints on how many data-driven ideas can be communicated per sentence, constraints on variability in word choice, and limited capabilities of analyzing data sets to determine the content that should be presented to a reader.

[0004] As technical solutions to these technical problems in the NLG arts, the inventors note that the assignee of the subject patent application has previously developed and commercialized pioneering technology that robustly generates narrative stories from data, of which a commercial embodiment is the QUILL™ narrative generation platform from Narrative Science Inc. of Chicago, IL. Aspects of this technology are described in the following patents and patent applications: U.S. Pat. Nos. 8,374,848, 8,355,903, 8,630,844, 8,688,434, 8,775,161, 8,843,363, 8,886,520, 8,892,417, 9,208,147, 9,251,134, 9,396,168, 9,576,009, 9,697,197, 9, 697,492, 9,720,890, 9,977,773, 10,185,477, 10,853,583, 11,144,838, 11,238,090, and 11,341,338; the entire disclosures of each of which are incorporated herein by reference.

[0005] The inventors have further extended on this pioneering work with improvements in AI technology as described herein.

[0006] For example, the inventors disclose how AI technology can be used in combination with composable communication goal statements and an ontology to facilitate a user's ability to quickly structure story outlines in a manner usable by a narrative generation system without any need to directly author computer code.

[0007] Moreover, the inventors also disclose that the ontology used by the narrative generation system can be built concurrently with the user composing communication goal statements. Further still, expressions can be attached to objects within the ontology for use by the narrative generation process when expressing concepts from the ontology as text in a narrative story. As such, the ontology becomes a re-usable and shareable knowledge-base for a domain that can be used to generate a wide array of stories in the domain by a wide array of users/authors.

[0008] The inventors further disclose techniques for editing narrative stories whereby a user's editing of text in the narrative story that has been automatically generated can in turn automatically result in modifications to the ontology and/or a story outline from which the narrative story was generated. Through this feature, the ontology and/or story outline is able to learn from the user's edits and the user is alleviated from the burden of making further corresponding edits of the ontology and/or story outline.

[0009] The inventors further disclose how the narrative analytics that are linked to communication goal statements can employ a conditional outcome framework that allows the content and structure of resulting narratives to intelligently adapt as a function of the nature of the data under consideration.

[0010] Further still, the inventors also disclose how “analyze” communication goals can be supported by the system, including various examples of communication goal statements that drive the generation of narratives that express various ideas that are deemed relevant to a given analysis communication goal.

[0011] Through these and other features, example embodiments of the invention provide significant technical advances in the NLG arts by harnessing AI computing to improve how narrative stories are generated from data sets while alleviating users from a need to directly code and re-code the narrative generation system, thereby opening up use of the A1-based narrative generation system to a much wider base of users (e.g., including users who do not have specialized programming knowledge).

Description

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] FIGS. **1**A-B and **2** depicts various process flows for example embodiments.

[0013] FIG. **3**A depicts an example process flow for composing a communication goal statement.

[0014] FIG. **3**B depicts an example ontology.

[0015] FIG. **3**C depicts an example process flow for composing a communication goal statement while also building an ontology.

[0016] FIG. **3**D depict an example of how communication goal statements can relate to an ontology and program code for execution by a process as part of a narrative generation process.

[0017] FIG. **4**A depicts examples of base communication goal statements.

[0018] FIG. **4**B depicts examples of parameterized communication goal statements corresponding to the base communication goal statements of FIG. **4**A.

[0019] FIG. **5** depicts a narrative generation platform in accordance with an example embodiment.

[0020] FIGS. **6**A-D depict a high level view of an example embodiment of a platform in accordance with the design of FIG. **5**.

[0021] FIG. **7** depicts an example embodiment of an analysis component of FIG. **6**C.

[0022] FIGS. **8**A-H depict example embodiments for use in an NLG component of FIG. **6**D.

[0023] FIG. **9** depicts an example process flow for parameterizing an attribute.

[0024] FIG. **10** depicts an example process flow for parameterizing a characterization.

[0025] FIG. **11** depicts an example process flow for parameterizing an entity type.

[0026] FIG. **12** depicts an example process flow for parameterizing a timeframe.

[0027] FIG. **13** depicts an example process flow for parameterizing a timeframe interval.
[0028] FIGS. **14**A-D illustrate an example of how a communication goal statement can include subgoals that drive the narrative generation process.
[0029] FIG. **15**A depicts an example conditional outcome data structure linked with one or more idea data structures.
[0030] FIG. **15**B depicts an example of narrative analytics that employ a conditional outcome framework to determine ideas to be expressed in a narrative.
[0031] FIG. **16** depicts an example embodiment for a conditional outcome framework that can be used by the narrative analytics associated with a communication goal statement for “Analyze Entity Group by Attribute”.
[0032] FIGS. **17**A and **17**B depict examples of how ideas can be linked to and delinked from outcomes within a conditional outcome framework in response to user input.
[0033] FIGS. **18**A and **18**B depict examples of narratives that can be generated using the conditional outcome framework of FIG. **16**.
[0034] FIGS. **19**A and **19**B depict an example embodiment for a conditional outcome framework that can be used by the narrative analytics associated with a communication goal statement for “Analyze Entity Group by Attribute **1** and Attribute **2**” and examples of narrative stories that can be generated thereby.
[0035] FIG. **20**A depicts an example embodiment for a conditional outcome framework that can be used by the narrative analytics associated with a communication goal statement for “Analyze Entity Group by a Change in Attribute (Over Time)” and an example of a narrative story that can be generated thereby.
[0036] FIGS. **20**B-D depict another example embodiment for a conditional outcome framework that can be used by the narrative analytics associated with a communication goal statement for “Analyze Entity Group by a Change in Attribute (Over Time)” and examples of a narrative stories that can be generated thereby.
[0037] FIGS. **21**A and **21**B depict an example embodiment for a conditional outcome framework that can be used by the narrative analytics associated with a communication goal statement for “Analyze Entity Group by Characterization” and examples of narrative stories that can be generated thereby.
[0038] FIGS. **22-293** illustrate example user interfaces for using an example embodiment to support narrative generation through composable communication goal statements and ontologies.

DETAILED DESCRIPTION OF EXAMPLE EMBODIMENTS

[0039] The example embodiments described herein further extend and innovate on the pioneering work described in the above-referenced and incorporated patents U.S. Pat. Nos. 9,576,009, 9,697,197, 9,697,492, 9,720,890, and 9,977,773, where explicit representations of communication goals are used by AI technology to improve how NLG technology generates narratives from data. With example embodiments described herein, AI technology is able to process a communication goal statement in relation to a data set in order to automatically generate narrative text about that data set such that the narrative text satisfies a communication goal corresponding to the communication goal statement. Furthermore, innovative techniques are disclosed that allow users to compose such communication goal statements in a manner where the composed communication goal statements exhibit a structure that promotes re-usability and robust story generation.
[0040] FIG. **1**A depicts a process flow for an example embodiment. At step **100**, a processor selects and parameterizes a communication goal statement. The processor can perform this step in response to user input as discussed below with respect to example embodiments. The communication goal statement can be expressed as natural language text, preferably as an operator in combination with one or more parameters, as elaborated upon below.
[0041] At step **102**, a processor maps data within the data set to the parameters of the communication goal statement. The processor can also perform this step in response to user input

as discussed below with respect to example embodiments.

[0042] At step **104**, a processor performs NLG on the parameterized communication goal statement and the mapped data. The end result of step **104** is the generation of narrative text based on the data set, where the content and structure of the narrative text satisfies a communication goal corresponding to the parameterized communication goal statement.

[0043] While FIG. **1**A describes a process flow that operates on a communication goal statement, it should be understood that multiple communication goal statements can be composed and arranged to create sections of an outline for a story that is meant to satisfy multiple communication goals. FIG. **1**B depicts an example process flow for narrative generation based on multiple communication goal statements. At step **110**, multiple communication goal statements are selected and parameterized to create sections of a story outline. At step **112**, a processor maps data within a data set to these communication goal statements as with step **102** (but for multiple communication goal statements). Step **114** is likewise performed in a manner similar to that of step **104** but on the multiple communication goal statements and the mapped data associated therewith. The end result of step **114** is a narrative story about the data set that conveys information about the data set in a manner that satisfies the story outline and associated communication goals.

[0044] It should be understood that steps **102** and **104**, as well as steps **112** and **114**, need not be performed in lockstep order with each other where step **102** (or **112**) maps all of the data before the system progresses to step **104** (or step **114**). These steps can be performed in a more iterative manner if desired, where a portion of the data is mapped at step **102** (or step **112**), followed by execution of step **104** (or step **114**) on that mapped data, whereupon the system returns to step **102**/**112** to map more data for subsequent execution of step **104**/**114**, and so on.

[0045] Furthermore, it should be understood that a system that executes the process flows of FIGS. **1**A and/or **1**B may involve multiple levels of parameterization. For example, not only is there parameterization in the communication goals to build story outlines, but there can also be parameterization of the resulting story outline with the actual data used to generate a story, as explained hereinafter with respect to example embodiments.

[0046] FIG. **2** depicts an example process flow that shows how a story outline can be composed as part of step **110**. The process flow of FIG. **2** can be performed by a processor in response to user input through a user interface. To begin the process, a name is provided for a section (step **120**). Within this section, step **100** is performed to define a communication goal statement for the subject section. At step **122**, the section is updated to include this communication goal statement. The process flow then determines whether another communication goal statement is to be added to the subject section (step **124**). If so, the process flow returns to steps **100** and **122**. If not, the process flow proceeds to step **126**. At step **126**, the process flow determines whether another section is to be added to the story outline. If so, the process flow returns to step **120**. Otherwise, the process flow concludes and the story outline is completed. Thus, through execution of the process flow of FIG. **2**, a processor can generate a story outline comprising a plurality of different sections, where each section comprises one or more communication goal statements. This story outline in turn defines the organization and structure of a narrative story generated from a data set and determines the processes required to generate such a story.

[0047] The previous example shows how an outline can be built by adding sections and parameterizing goals completely from scratch. The user is generally not expected to start from scratch, however. A narrative generation system instance will generally include a library of prebuilt components that users can utilize to more easily and quickly build out their outline. The narrative generation system's library provides access to previously parameterized and composed goals, subsections, sections, and even fully defined outlines. These re-usable components come fully parameterized, but can be updated or adjusted for the specific project. These changes are initially isolated from the shared library of components.

[0048] Components from the system's shared library can be used in two ways. First, a new project can be created from an entire project blueprint providing all aspects of a project already defined. This includes sample data, data views, the ontology, outline, sections, parameterized goals, and data mappings. Second, a user can pull in predefined components from the system's library ad hoc while building a new project. For example, when adding a section to an outline, the user can either start from scratch with an empty section or use a predefined section that includes a set of fully parameterized goals.

[0049] The system's library of components can be expanded by users of the platform through a mechanism that enables users to share components they have built. Once a component (outline, ontology, section, etc.) is shared, other users can then use them from the system's library in their own projects.

Composable Communication Goal Statements:

[0050] FIG. **3**A depicts an example process flow for composing a communication goal statement, where the process flow of FIG. **3**A can be used to perform step **100** of FIGS. **1**A and **2** (see also step **110** of FIG. **1**B). The process flow of FIG. **3**A can be performed by a processor in response to user input through a user interface. The process flow begins at step **300** when the processor receives user input that indicates a base communication goal statement. The base communication goal statement serves as a skeleton for a parameterized and composed communication goal and may comprise one or more base goal elements that serve to comprise the parameterized and composed communication goal statement. Base goal elements are the smallest composable building blocks of the system out of which fully parameterized communication goal statements are constructed. Internal to the system, they are structured objects carrying necessary information to serve as the placeholders for parameters that are to be determined during the composition process. Communication goal statements are displayed to the user in plain language describing the goal's operation and bound parameters. In an example embodiment, the base communication goal statement is represented to a user as an operator and one or more words, both expressed in natural language, and where operator serves to identify a communication goal associated with the base communication goal statement and where the one or more words stand for the base goal elements that constitute parameters of the parameterized communication goal statement. FIG. **4**A depicts examples of base communication goal statements as presented to a user that can be supported by an example embodiment.

[0051] As shown by FIG. **4**A, base communication goal statement **402** is “Present the Value” where the word “Present” serves as the operator **410** and “Value” serves as the parameter placeholder **412**. The operator **410** can be associated with a set of narrative analytics (discussed below) that define how the AI will analyze a data set to determine the content that is to be addressed by a narrative story that satisfies the “Present the Value” communication goal. The parameter placeholder **412** is a field through which a user specifies an attribute of an entity type to thereby define a parameter to be used as part of the communication goal statement and subsequent story generation process. As explained below, the process of parameterizing the parameter placeholders in the base communication goal statements can build and/or leverage an ontology that represents a knowledge base for the domain of the story generation process. As shown by FIG. **4**B, another example of a base communication goal statement is base communication goal statement **404**, which is expressed as “Present the Characterization”, but could also be expressed as “Characterize the Entity”. In these examples, “Present” (or “Characterize”) can serve as operator **414** can “Characterization” (or Entity”) can serve as a parameter placeholder **416**. This base communication goal statement can be used to formulate a communication goal statement geared toward analyzing a data set in order to express an editorial judgment about data within the data set.

[0052] As shown by FIG. **4**B, another example of a base communication goal statement is base communication goal statement **406**, which is expressed as “Compare the Value to the Other Value”, where “Compare” serves as operator **418**, “Value” serves as a parameter placeholder **420**, and “Other Value” serves as parameter placeholder **422**. The “Compare” operator **418** can be associated with a set of narrative analytics that are configured to compute various metrics indicative of a comparison between the values corresponding to specified attributes of specified entities to support the generation of a narrative that expresses how the two values compare with each other.

[0053] Another example of a base communication goal statement is “Callout the Entity” **408** as shown by FIG. **4**A. In this example, “Callout” is operator **424** and “Entity” is the parameter placeholder **426**. The “Callout” operator **424** can be associated with a set of narrative analytics that are configured to compute various metrics by which to identify one or more entities that meet a set of conditions to support the generation of a narrative that identifies such an entity or entities in the context of these conditions.

[0054] It should be understood that the base communication goal statements shown by FIG. **4**A are just examples, and a practitioner may choose to employ more, fewer, or different base communication goal statements in a narrative generation system. For example, additional base communication goal statements could be employed that include operators such as “Review”, “Analyze”, “Explain”, “Predict” etc. to support communication goal statements associated with communication goals targeted toward such operators. An example structure for a base “Review” communication goal statement could be “Review the [timeframe interval] [attribute] of [the entity] over [timeframe]”. An example structure for a base “Explain” communication goal statement could be “Explain the [computed attribute] of [the entity] in [a timeframe]”. Also, example embodiments describing how communication goal statements with an “Analyze” operator can be used to support the generation of narratives that satisfy an “analysis” communication goal are discussed below.

[0055] The system can store data representative of a set of available base communication goal statements in a memory for use as a library. A user can then select from among this set of base communication goal statements in any of a number of ways. For example, the set of available base communication goal statements can be presented as a menu (e.g., a drop down menu) from which the user makes a selection. As another example, a user can be permitted to enter text in a text entry box. Software can detect the words being entered by the user and attempt to match those words with one of the base communication goal statements as would be done with auto-suggestion text editing programs. Thus, as a user begins typing the character string “Compa . . . ” the software can match this text entry with the base communication goal statement of “Compare the Value to the Other Value” and select this base communication goal statement at step **300**.

[0056] Returning to FIG. **3**A, the process flow at steps **302-306** operates to parameterize the base communication goal statement by specifying parameters to be used in place of the parameter placeholders in the base communication goal statement. One of the technical innovations disclosed by the inventors is the use of an ontology **320** to aid this part of composing the communication goal statement. The ontology **320** is a data structure that identifies the types of entities that exist within the knowledge domain used by the narrative generation system to generate narrative stories in coordination with communication goal statements. The ontology also identifies additional characteristics relating to the entity types such as various attributes of the different entity types, relationships between entity types, and the like.

[0057] Step **302** allows a user to use the existing ontology to support parameterization of a base communication goal statement. For example, if the ontology **320** includes an entity type of “Salesperson” that has an attribute of “Sales”, a user who is parameterizing base communication goal statement **402** can cause the processor to access the existing ontology **320** at step **304** to select “Sales of the Salesperson” from the ontology **320** at step **306** to thereby specify the parameter to be used in place of parameter placeholder **412** and thereby create a communication goal statement of “Present the Sales of the Salesperson”.

[0058] Also, if the existing ontology **320** does not include the parameters desired by a user, step **306** can operate by a user providing user input that defines the parameter(s) to be used for parameterizing the communication goal statement. In this situation, the processor in turn builds/updates the ontology **320** to add the parameter(s) provided by the user. For example, if the ontology **320** did not already include “Sales” as an attribute of the entity type “Salesperson”, steps **306-308** can operate to add a Sales attribute to the Salesperson entity type, thereby adapting the ontology **320** at the same time that the user is composing the communication goal statement. This is a powerful innovation in the art that provides significant improvement with respect to how artificial intelligence can learn and adapt to the knowledge base desired by the user for use by the narrative generation system.

[0059] At step **310**, the processor checks whether the communication goal statement has been completed. If so, the process flow ends, and the user has composed a complete communication goal statement. However, if other parameters still need to be specified, the process flow can return to step **302**. For example, to compose a communication goal statement from the base communication goal statement **406** of “Compare the Value to the Other Value”, two passes through steps **302-308** may be needed for the user to specify the parameters for use as the Value and the Other Value.

[0060] FIG. **4**B shows examples of parameterized communication goal statements that can be created as a result of the FIG. **3**A process flow. For example, the base communication goal statement **402** of FIG. **4**A can be parameterized as communication goal statement **402** (“Present the Price of the Car”, where the parameter placeholder **412** has been parameterized as parameter **412***b*, namely “Price of the Car” in this instance, with “Price” being the specified attribute of a “Car” entity type). Similarly, the base communication goal statement **402** of FIG. **4**A could also be parameterized as “Present the Average Value of the Deals of the Salesperson”, where the parameter placeholder **412** has been parameterized as parameter **412***b*, namely “Average Value of the Deals of the Salesperson” in this instance).

[0061] FIG. **4**B also shows examples of how base communication goal statement **404** can be parameterized (see relatively lengthy “Present the Characterization of the Highest Ranking Department in the City by Expenses in terms of the Difference Between its Budget and Expenses” statement **404***b***1** where the specified parameter **404***b***1** is the “Characterization of the Highest Ranking Department in the City by Expenses in terms of the Difference Between its Budget and Expenses”; see also its substantially equivalent in the form of statement **404***b***2**).

[0062] Also shown by FIG. **4**B are examples of parameterization of base communication goal statement **406**. A first example is the communication goal statement **406***b* of “Compare the Sales of the Salesperson to the Benchmark of the Salesperson” where the specified parameter for “Value” **420** is “Sales of the Salesperson” **420***b* and the specified parameter for “Other Value” **422** is “Benchmark of the Salesperson” **422***b*. A second example is the communication goal statement **406***b* of “Compare the Revenue of the Business to the Expenses of the Business” where the specified parameter for “Value” **420** is “Revenue of the Business” **420***b* and the specified parameter for “Other Value” **422** is “Expenses of the Business” **422***b*.

[0063] Also shown by FIG. **4**B are examples of parameterization of base communication goal statement **408**. A first example is the communication goal statement **408***b* of “Callout the Highest Ranked Salesperson by Sales” where the specified parameter for “Entity” **426** is the “Highest Ranked Salesperson by Sales” **426***b*. A second example is the communication goal statement **408***b* of “Callout the Players on the Winning Team” where the specified parameter for “Entity” **426** is “Players on the Winning Team” **426***b*. A third example is the communication goal statement **408***b* of “Callout the Franchises with More than $1000 in Daily Sales” where the specified parameter for “Entity” **426** is “Franchises with More than $1000 in Daily Sales” **426***b*.

[0064] As with the base communication goal statements, it should be understood that a practitioner may choose to employ more, fewer, or different parameterized communication goal statements in a narrative generation system. For example, a parameterized Review communication goal statement could be “Review the weekly cash balance of the company over the year”, and a parameterized Explain communication goal statement could be “Explain the profit of the store in the month”.

Ontology Data Structure:

[0065] FIG. **3**B depicts an example structure for ontology **320**. The ontology **320** may comprise one or more entity types **322**. Each entity type **322** is a data structure associated with an entity type and comprises data that describes the associated entity type. An example of an entity type **322** would be a “salesperson” or a “city”. Each entity type **322** comprises metadata that describes the subject entity type such as a type **324** (to identify whether the subject entity type is, e.g., a person, place or thing) and a name **326** (e.g., “salesperson”, “city”, etc.). Each entity type **322** also comprises one or more attributes **330**. For example, an attribute **330** of a “salesperson” might be the “sales” achieved by a salesperson. Additional attributes of a salesperson might be the salesperson's gender and sales territory.

[0066] Attributes **330** can be represented by their own data structures within the ontology and can take the form of a direct attribute **330***a* and a computed value attribute **330***b*. A direct attribute **330***a* is an attribute of an entity type that can be found directly within a data set (e.g., for a data set that comprises a table of salespeople within a company where the salespeople are identified in rows and where the columns comprise data values for information such as the sales and sales territory for each salesperson, the attribute “sales” would be a direct attribute of the salesperson entity type because sales data values can be found directly within the data set). A computed value attribute **330***b* is an attribute of an entity type that is derived in some fashion from the data set. Continuing with the example above, a direct attribute for the salesperson entity type might be a percentage of the company's overall sales that were made by the salesperson. This information is not directly present in the data set but instead is computed from data within the data set (e.g., by summing the sales for all salespeople in the table and computing the percentage of the overall sales made by an individual salesperson).

[0067] Both the direct attributes **330***a* and computed value attributes **330***b* can be associated with metadata such as a type **340** (e.g., currency, date, decimal, integer, percentage, string, etc.), and a name **342**. However, computed value attributes **330***b* can also include metadata that specifies how the computed value attribute is computed (a computation specification **348**). For example, if a computed value attribute **330***b* is an average value, the computation specification **348** can be a specification of the formula and parameters needed to compute this average value.

[0068] Each entity type **322** may also comprise one or more characterizations **332**. For example, a characterization **332** of a “salesperson” might be a characterization of how well the salesperson has performed in terms of sales (e.g., a good performer, an average performer, a poor performer). Characterizations can be represented by their own data structures **332** within the ontology. A characterization **332** can include metadata such as a name **360** (e.g., sales performance). Also, each characterization **332** can include a specification of the qualifications **364** corresponding to the characterization. These qualifications **364** can specify one or more of the following: (1) one or more attributes **330** by which the characterization will be determined, (2) one or more operators **366** by which the characterization will be determined, and (3) one or more value(s) **368** by which the characterization will be determined. For example, a “good performer” characterization for a salesperson can be associated with a qualification that requires the sales for the salesperson to exceed a defined threshold. With such an example, the qualifications **364** can take the form of a specified attribute **330** of “sales”, an operator **366** of “greater than”, and a value **368** that equals the defined threshold (e.g., $100,000).

[0069] Each entity type **322** may also comprise one or more relationships **334**. Relationships **334** are a way of identifying that a relationship exists between different entity types and defining how those different entity types relate to each other. Relationships can be represented by their own data structures **334** within the ontology. A relationship **334** can include metadata such as the related entity type **350** with respect to the subject entity type **322**. For example, a “salesperson” entity type can have a relationship with a “company” entity type to reflect that the salesperson entity type belongs to a company entity type. The ontological objects (e.g., entity types **322**, direct attributes **330***a*, computed value attributes **330***b*, characterizations **332**, and relationships **334**) may also comprise data that represents one or more expressions that can be used to control how the corresponding ontological objects are described in narrative text produced by the narrative generation system.

[0070] For example, the entity type **322** can be tied to one or more expressions **328**. When the narrative generation process determines that the subject entity type needs to be described in narrative text, the system can access the expression(s) **328** associated with the subject entity type to determine how that entity type will be expressed in the narrative text. The expression(s) **328** can be a generic expression for the entity type **322** (e.g., the name **326** for the entity type, such as the name “salesperson” for a salesperson entity type), but it should be understood that the expression(s) **32** may also or alternatively include alternate generic names (e.g., “sales associate”) and specific expressions. By way of example, a specific expression for the salesperson entity type might be the name of a salesperson. Thus, a narrative text that describes how well a specific salesperson performed can identify the salesperson by his or her name rather than the more general “salesperson”. To accomplish this, the expression 328 for the salesperson can be specified indirectly via a reference to a data field in a data set (e.g., if the data set comprises a table that lists sales data for various sales people, the expression 328 can identify a column in the table that identifies each salesperson's name). The expression(s) **328** can also define how the subject entity type will be expressed when referring to the subject entity type as a singular noun, as a plural noun, and as a pronoun.

[0071] The expression(s) **346** for the direct attributes **330***a* and computed value attributes **330***b* can take a similar form as and operate in a manner similar to the expression(s) for the entity types **322**; likewise for the expression(s) **362** tied to characterizations **332** (although it is expected that the expressions **362** will often include adjectives and/or adverbs in order to better express the characterization **332** corresponding to the subject entity type **322**). The expression(s) **352** for relationships **334** can describe the nature of the relationship between the related entity types so that this relationship can be accurately expressed in narrative text if necessary. The expressions **352** can typically take forms such as “within” (e.g., a “city” entity type within a “state” entity type, “belongs to” (e.g., a “house” entity type that belongs to a “person” entity type, “is employed by” (a “salesperson” entity type who is employed by a “company” entity type), etc.

[0072] Another ontological object can be a timeframe **344**. In the example of FIG. **3**B, timeframes **344** can be tied to direct attributes **330***a* and/or computed value attributes **330***b*. A direct attribute **330***a* and/or a computed value attribute **330***b* can either be time-independent or time-dependent. A timeframe **344** can define the time-dependent nature of a time-dependent attribute. An example of a time-dependent attribute would be sales by a salesperson with respect to a data set that identifies each salesperson's sales during each month of the year. The timeframe **344** may comprise a timeframe type **356** (e.g., year, month, quarter, hour, etc.) and one or more expressions(s) **358** that control how the subject timeframe would be described in resultant narrative text. Thus, via the timeframe **344**, a user can specify a timeframe parameter in a communication goal statement that can be used, in combination with the ontology **320**, to define a specific subset of data within a data set for consideration. While the example of FIG. **3**B shows timeframes **344** being tied to direct attributes **330***a* and computed value attributes **330***b*, it should be understood that a practitioner might choose to make timeframes **344** only attachable to direct attributes **330***a*. Also, a practitioner might choose to make timeframes **344** also applicable to other ontological objects, such as characterizations **332**, entity types **322**, and/or even relationships **334**. As indicated in connection with FIG. **3**A, users can create and update the ontology **320** while composing communication goal statements. An example embodiment for such an ability to simultaneously compose communication goal statements and build/update an ontology is shown by FIG. **3**C. At step **370**, the system receives a text string entry from a user (e.g., through a text entry box in a user interface (UI)). As indicated, this text entry can be a natural language text entry to facilitate ease of use by users. Alternative user interface models such as drag and drop graphical user interfaces or structured fill in the blank templates could also be used for this purpose.

[0073] At step **372**, the processor attempts to match the received text string to a base communication goal statement that is a member of a base communication goal statement library **504** (see FIG. **4**A). This matching process can be a character-based matching process where the processor seeks to find a match on an ongoing basis as the user types the text string. Thus, as a user types the string "Comp", the processor may be able to match the text entry to the "Compare the Value to the Other Value" base communication goal statement. Based on this matching, the system can auto-fill or auto-suggest a base communication goal statement that matches up with the received text entry (step **374**). At this point, the system can use the base communication goal statement as a framework for guiding the user to complete the parameterization of the communication goal statement.

[0074] At step **376**, the system continues to receive text string entry from the user. At step **378**, the processor attempts to match the text string entry to an object in ontology **320**. Is there is a match (or multiple matches), the system can present a list of matching ontological objects for user selection (step **380**). In this fashion, the system can guide the user to define parameters for the communication goal statement in terms of objects known within ontology **320**. However, if the text string does not match any ontological objects, the system can provide the user with an ability to create a new object for inclusion in the ontology (steps **382-384**). At step **382**, the system provides the user with one or more UIs through which the user creates object(s) for inclusion in ontology **320** (e.g., defining an entity type, attribute, characterization, relationship, and/or timeframe). At step **384**, the system receives the user input through the UI(s) that define the ontological objects. The ontology can thus be updated at step **308** in view of the text string entered by a user that defines a parameter for the communication goal statement.

[0075] If step **310** results in a determination that the communication goal statement has not been completed, the process flow returns to step **376** as the user continues entering text. Otherwise, the process flow concludes after step **310** if the communication goal statement has been fully parameterized (see FIG. **4**B for examples of parameterized communication goal statements).

[0076] Through the use of composable communication goal statements and ontology **320**, example embodiments are capable of generating a robust array of narrative stories about data sets that satisfy flexibly-defined communication goals without requiring a user to directly author any program code. That is, a user need not have any knowledge of programming languages and does not need to write any executable code (such as source code) in order to control how the narrative generation platform automatically generates narrative stories about data sets. To the extent that any program code is manipulated as a result of the user's actions, such manipulation is done indirectly as a result of the user's higher level compositions and selections through a front end presentation layer that are distinct from authoring or directly editing program code. Communication goal statements can be composed via an interface that presents them in natural language as disclosed herein, and ontologies can similarly be created using intuitive user interfaces that do not require direct code writing. FIG. **3**D illustrates this aspect of the innovative design. In an example embodiment, communication goal statements **390** (e.g., **3901** and **3902**) are composed by a user using an interface that presents the base goal elements as natural language text where one or more words represent the goal operators and one or more words serve to represent the parameters as discussed above. These parameters, in turn, map into ontology **320** and thus provide the constraints necessary for the narrative generation platform to appropriately determine how to analyze a data set and generate the desired narrative text about the data set (described in greater detail below). Hidden from the user are code-level details. For example, a computed value attribute (such as **330***b*.sub.n) is associated with parameterized computational logic **394** that will be executed to compute its corresponding computed value attribute. Thus, if the computed value attribute **330***b*.sub.n is an average value of a set of data values, the computational logic **394** can be configured to (1) receive a specification of the data values as input parameters, (2) apply these data values to a programmed formula that computes an average value, and (3) return the computed average value as the average value attribute for use by the narrative generation platform. As another example, computational logic **392** and **396** can be configured to test qualifications for corresponding characterizations **3321** and **3322** respectively. The data needed to test the defined qualifications can be passed into the computational logic as input parameters, and the computational logic can perform the defined qualification tests and return an identification of the determined characterization for use by the narrative generation platform. Similar computational logic structures can leverage parameterization and the ontology **320** to perform other computations that are needed by the narrative generation platform.

[0077] The inventors also disclose that the ontology **320** can be re-used and shared to generate narrative stories for a wide array of users. For example, an ontology **320** can be built that supports generation of narrative stories about the performance of retail businesses. This ontology can be re-used and shared with multiple users (e.g., users who may have a need to generate performance reports for different retail businesses). Accordingly, as ontologies **320** are created for different domains, the inventors envision that technical value exists in maintaining a library of ontologies **320** that can be selectively used, re-used, and shared by multiple parties across several domains to support robust narrative story generation in accordance with user-defined communication goals.

Example Narrative Generation Architecture Using Composed Communication Goal Statements:

[0078] FIG. **5** depicts a narrative generation platform in accordance with an example embodiment. An example embodiment of the narrative generation platform can include two artificial intelligence (AI) components. A first AI component **502** can be configured to determine the content that should be expressed in a narrative story based on a communication goal statement (which can be referred to as “what to say” AI **502**). A second AI component **504** can be configured to perform natural language generation (NLG) on the output of the first AI component **502** to produce the narrative story that satisfies the communication goal statement (where the AI component **504** can be referred to as “how to say it” AI **504**).

[0079] The platform can also include a front end presentation layer **570** through which user inputs **572** are received to define the composed communication goal statement **390**. This presentation layer **570** can be configured to allow user composition of the communication goal statement **390** using natural language inputs. As mentioned herein, it can also employ structured menus and/or drag/drop features for selecting elements of a communication goal statement. Examples of various user interfaces that can be used by the presentation layer **570** are shown in FIGS. **22-293** and further described in Appendix A. As can be seen from these sample UIs, the presentation layer **570** can also leverage the ontology **320** and source data **540** to facilitate its user interactions.

[0080] The “what to say” AI **502** can be comprised of computer-executable code resident on a non-transitory computer-readable storage medium such as computer memory. The computer memory may be distributed across multiple memory devices. One or more processors execute the computer code in cooperation with the computer memory. AI **502** operates on a composed communication goal statement **390** and ontology **320** to generate a computed story outline **528**. AI **502** includes a communication goal statement interpreter **506**, which is configured to process and interpret the communication goal statement **390** to select a set of narrative analytics that are to be used to analyze a data set about which the narrative story will be generated. The computer memory may include a library **508** of narrative analytics **510** (e.g., **510**.sub.1, **510**.sub.2, **510**.sub.3, . . .). The narrative analytics **510** may take the form of parameterized computer code that performs analytical operations on the data set in order to facilitate a determination as to what content should be included in the narrative story so that the communication goal(s) corresponding to the communication goal statement **390** are satisfied. Examples of narrative analytics **510** can be the computational logic **392**, **394**, and **396** shown in FIG. **3**D.

[0081] AI **502** can maintain a mapping that associates the various operators that may be present in communication goal statements (e.g., “Present”, “Compare”, etc.) to a sequence or set of narrative analytics that are to be performed on data in order to support the data analysis needed by the platform to generate narrative stories that satisfy the communication goal statement **390**. Thus, the "Compare" operator can be associated with a set of narrative analytics that do simple difference (a−b), absolute difference (abs (a−b)), or percent difference ((b−a)/b). In an example embodiment, the mapping can also be based on the parameters that are included in the communication goal statement **390**. The mapping can take the form of a data structure (such as a table) that associates operators (and possibly also parameters) with sets of narrative analytics **510** from library **508**. Interpreter **506** can then read and interpret the communication goal statement **390** to identify the operator included in the communication goal statement, access the mapping data structure to map the identified operator to its corresponding set of narrative analytics **510**, and select the mapped narrative analytics. These selected narrative analytics **512** in turn drive downstream operations in AI **502**.

[0082] AI **502** can also include computer code **516** that is configured to determine the data requirements that are needed by system to generate a narrative story in view of the selected narrative analytics **512** and the parameters that are included in the communication goal statement **390**. This code **516** can walk through the selected narrative analytics **512**, the communication goal statement **390**, and ontology **320** to identify any parameters and data values that are needed during execution of the selected narrative analytics **512**. For example, the communication goal statement **390** may include parameters that recite a characterization of an entity. Computer code **390** can identify this characterization in the communication goal statement and access the ontology **320** to identify the data needed to evaluate the characterization of the subject entity such as the attribute(s) **330** and value(s) **368** needed for the subject characterization **332** in ontology **320**. The ontology **320** can then be further parsed to determine the data requirements for the subject attribute(s) needed by the subject characterization **332**, and so on until all data requirements for the communication goal statement **390** and selected narrative analytics **512** are determined. This ultimately yields a set of data requirements **518** that define the data needed by AI **502** in order to support the data analysis used to determine the content to be expressed in the narrative story. In situations where the input to AI **502** comprises multiple communication goal statements **390** in a story outline, code **516** can be configured to walk through the outline to assemble a list of the data requirements for all of the communication goal statements in the outline.

[0083] Once the data requirements **518** have been determined, the AI **502** can execute computer code **522** that maps those data requirements **522** to source data **540**. (This can be done either in a "batch" model wherein all the data requirements are determined first, and the code to map those to source data is executed; or it can be done individually for each data requirement either as needed or as the other information necessary to make the determination becomes available.) The source data **540** serves as the data set from which the narrative story will be generated. Source data **540** can take the form of data in a database, data in spreadsheet files, or other structured data accessible to AI **502**. Computer code **522** can use a data structure **520** (such as a table) that associates parameters from the data requirements to parameters in the source data to perform this mapping. For example, consider a scenario where the communication goal statement is "Present the Sales of the Salesperson". The data requirements **518** for this communication goal statement may include a parameter that corresponds to the "sales" attribute of a salesperson. The source data **540** may include a data table where a column labeled as "Amount Sold ($)" identifies the sales amount for each salesperson in a company. The parameter mapping data structure **520** can associate the "Sales" parameter from the data requirements **518** to the "Amount Sold ($)" column in the source data **540** so that AI **502** accesses the proper data. This parameter mapping data structure **520** can be defined by an author when setting up the system, as discussed hereinafter. The output of computer code **522** can be a set of mapped source data **524** for use by the selected narrative analytics **512**.

[0084] Computer code **522** can also map data requirements to source data using story variable(s) **542**. For example, the communication goal statement **390** might be "Compare the Sales of Salesperson "John Smith" to the Benchmark of the Salesperson". The mapped source data **524** can identify where in the source data the sales and benchmark for salespeople can be found. If the source data **540** includes sales data for multiple salespeople (e.g., rows in a data table correspond to different sales people while columns in the data table correspond to sales amounts and benchmarks for salespeople), the selection of a particular salesperson can be left as a story variable **542** such that the parameter mapping data structure **520** does not identify which specific row to use as the salesperson and instead identifies the salesperson data requirement as a story variable. When a user composes the communication goal statement such that "John Smith" is expressed in the statement where the salesperson parameter is located, the computer code **522** can use "John Smith" in the communication goal statement **390** as the story variable **542** that governs the selection of which row of source data **540** should be used. Similarly, the benchmark parameter might be expressed as a story variable **542**. For example, the source data **540** may not include a benchmark field, but the composed communication goal statement might express a number to be used as the benchmark. In such a situation, this number could be a story variable **542** used by the system.

[0085] FIGS. **41** and **220-232**, described below with reference to Appendix A, depict example GUIs through which a user can map the determined data requirements for a story outline to source data and story variables. These GUIs can be configured to list each data requirement in association with a user input mechanism through which the user can identify where in the source data a data requirement can be found (and whether a data requirement is to be parameterized as a story variable). As explained in Appendix A with respect to an example embodiment, the source data can take a number of forms, such as tabular data and document-based data, and the data requirements GUIs can be configured to accommodate both types. FIGS. **233-250** and their supporting description in Appendix A further describe how source data can be managed in an example embodiment of the system.

[0086] AI **502** can also include computer code **526** that executes the selected narrative analytics **512** using the mapped source data **524** (and potentially any story variable(s) **542**) to produce a computed story outline **528**. The narrative analytics **512** specifies at least four components: the input parameters (e.g., an entity to be ranked, a metric it is to be ranked by, and a group in which it is to be ranked); the code that will execute the narrative analytics (i.e., that will determine the rank of the entity in the group according to the metric); the output parameters (i.e., the rank of the entity); and a statement form containing the appropriate input and output parameters that will form the appropriate statement for inclusion in the computed outline (in this case, rank (entity, metric, group, rankvalue)). The communication goal statement **390** can be associated with a general story outline that provides the basic structure for the narrative story to be generated. However, this general story outline will not be populated with any specific data—only general identifications of parameters. Through execution of the selected narrative analytics by computer code **526**, this general story outline can be populated with specific data in the form of the computed story outline **528**. For example, continuing with an example from above where the communication goal statement **390** is "Compare the Sales of Salesperson "John Smith" to the Benchmark of the Salesperson", the selected narrative analytics may include parameterized code that computes data indicative of the difference between John Smith's sales amount and the benchmark in both absolute terms (e.g., performing a subtraction between the sales amount and the benchmark) and as a percentage (e.g., dividing the subtracted difference by the benchmark and multiplying by **100**). Code **526** executes these narrative analytics to compute data values for use in the story outline. These data values are then embedded as values for the parameters in the appropriate statement forms associated with the narrative analytics to produce statements for inclusion in the computed outline. The statement will be included in the computed outline as a new element of the section containing the communication goal for which it was computed, under the node representing that communication goal. Code **526** will progress through the execution of the selected narrative analytics using mapped source data **524** and story variable(s) **542** (if any) until all elements of the story outline have been populated with statements. Also associated with communication goals are characterizations that serve to express a characterization or editorialization of the facts reported in the statements in a manner that may have more narrative impact that just a reporting of the facts themselves. For example, rather than saying that an entity is ranked first, we might say that it is the best. (In another approach, these might be associated with sections rather than communication goals.) The characterizations associated with each communication goal are assessed with respect to the statements generated by the narrative analytics in response to that goal. This results in generating additional propositions or statements corresponding to those characterizations for inclusion in the computed outline in those cases when the conditions for those characterizations are met by the input statements. The characterizations are also linked to the statements which they characterize. The result of this process is a computed story outline **528** that serves to identify the content that is to be expressed in the narrative story.

[0087] The “how to say it” AI **504** can be comprised of computer-executable code resident on a non-transitory computer-readable storage medium such as computer memory. The computer memory may be distributed across multiple memory devices. One or more processors execute the computer code in cooperation with the computer memory. AI **504** employs NLG logic **530** to generate a narrative story **550** from the computed story outline **528** and ontology **320**. As indicated above, objects in ontology **320** can be associated with expressions (e.g., expressions **328**, **346**, **352**, **358**, and **362**) that can be used by NLG **530** to facilitate decision-making regarding the appropriate manner of expressing the content in the computed story outline **528**. Thus, NLG **530** can access the ontology **320** when forming sentences from the computed story outline **528** for use in the narrative story **550**. Example embodiments of NLG **530** are discussed below with reference to FIGS. **6**D and **8**A-H.

[0088] Once again, by leveraging predefined sets of parameterized narrative analytics **510**, AI **502** is able to shield the low level program coding from users so that a user need only focus on composing communication goal statements **390** in a natural language in order to determine the content that is to be included in a narrative story. Further still, AI **504** also operates transparently to users so that a narrative story **550** can be generated from a composed communication goal statement **390** without requiring the user to directly write or edit program code.

Example Platform Operation:

[0089] FIG. **6**A depicts a high level view of an example embodiment of a platform in accordance with the design of FIG. **5**. The narrative generation can proceed through three basic stages: setup (an example of which is shown by FIG. **6**B), analysis (an example of which is shown by FIG. **6**C), and NLG (an example of which is shown by FIG. **6**D). The operation of the FIG. **6**A embodiment can be described in the context of a simple example where the project has an outline with a single section and a single communication goal statement in that section. The communication goal statement can be “Present the sales of the salesperson”. In this example, “salesperson” is an entity type in the ontology and it has an attribute of “sales”. Also, the project has a single data view backed by a static file that contains the names and sales data for the salespeople.

[0090] During setup, the system loads the story configuration from a configuration store. The configuration store is a database where configurations are maintained in persistent form, managed, and versioned. The configuration for a story includes items representing the outline (sections, communication goals, and their components), the ontology (entity types, relationships, timeframe types), and data connectors (sources, data mappings). Once the configuration for the story is loaded into memory, the story outline is constructed, as shown in FIG. **6**B. The story outline is a hierarchical organization of sections and communication goals (see FIG. **2**). At this time, along with constructing the story outline, the connectors to the data sources are initialized. These will be used as needed during the story generation process to access the necessary data required by the narrative analytics specified in the outline. Specifically how this is accomplished can depend on whether the data is passed in via an API, in a static file managed by the system, or via a connection to a database.

[0091] Once the setup phase is complete, the outline can be used to govern the generation of a story. This is accomplished by traversing the outline and executing the analytics associated with each communication goal statement; and the results serve to parameterize the associated statement forms of the communication goal in order to generate the facts of the story (see FIG. **6**C). These facts are then organized into the computed outline as described above.
[0092] When this generation process is invoked by a client, e.g., via an API request, the client provides certain values for parameters of the configuration. In this instance, for example, the story is about the sales of some particular salesperson. So the client may need to provide a unique identifier for the specific salesperson which can be interpreted via the mapping provided between parameters of the story outline and the data source to be used.
[0093] As shown by FIG. **7**, the narrative analytics can access source/customer data through Entity and Entity Collection objects. These objects provide an interface based on the project ontology **320** and hide the source of the data from other components. These objects can use Entity Types, mappings from relevant Attributes of the Entity Types to data sources and specifiers (e.g., columns or column names in tables or databases, or keypaths in documents, etc.) as previously specified by the user during configuration, and data interfaces to access the actual relevant data. Some computations that comprise aspects of the narrative analytics, such as sorting and certain aggregations, can be handled by the data stores themselves (e.g., as database operations). The specific Entity objects provide methods to invoke these external operations, such as parameterizable database queries.
[0094] Continuing with the example, the single communication goal statement in this case, “Present the Sales of the Salesperson”, is made up of two base communication goal statements, composed together by embedding one inside the other. The top level statement is AttributeOfEntity (AttributeName, <Entity>), and its Entity parameter is satisfied by the embedded statement EntityById (Id). EntityById is resolved first. This is computed by retrieving the entity's ID as provided by the client when invoking the generation process, e.g., via an API request. EntityById creates an (internal) Entity object corresponding to the (external) ID and returns that Entity object as its result. This internal Entity object is a new Entity of the appropriate Entity Type as specified in the configuration and with appropriate attributes as determined by the entity data mapping, in this instance, since we are talking about a Salesperson, relevant attributes of the Salesperson in question such as his or her name, gender, sales, office—whatever in fact the configuration specifies be retrieved or computed. This result is in the form of the embedded communication goal statement, namely, EntityById (Id, <Entity>); it is then, in turn, passed into the top-level AttributeOfEntity statement along with the attribute name “sales”. The AttributeOfEntity analytic comprises code that takes the entity object and returns the corresponding value for that attribute of the entity as its result. The analytic looks up where to get the attribute data based on the entity data mappings provided during configuration, and retrieves the specific relevant attribute data from the client's data. The results for both of these are wrapped up in statement forms to produce statements as described above, and these statements are then added to the Computed Outline. In this specific case, as mentioned above, the statements are composed by one being embedded inside the other. The resulting compound statement added to the Computed Outline in this instance, fully parameterized, would look something as follows: AttributeOfEntity (‘Sales’, EntityByID (‘1234’, Salesperson1234), 15000).
[0095] FIG. **6**D shows a high level view of NLG being performed on a computed outline in order to generate a narrative story. FIGS. **8**A-**8**H elaborate on this NLG process.
[0096] As shown by FIG. **8**A, the NLG process starts with the Computed Outline. Each phase of the NLG process walks through the Computed Outline and processes each computed statement form individually. Some stages look across multiple statements at once (such as Model Muting (see FIG. **8**B) and Entity Referencing (see FIG. **8**F), described below.
[0097] The first phase, Model Generation, converts the compound statements in the computed

outline into NLGModel graphs, as shown by FIG. **8**A. Model graphs are similar to the compound statement structures, but are structured specifically for constructing sentences. For example, dependencies between nodes in the model graph will represent where dependent clauses should be placed on the sentence. An NLGModel provides a mechanism for generating sentences, phrases, and words needed to produce a story. There is model type for each concept that needs to be expressed from authoring mapping to each individual type of statement included in the computed outline. Examples include attributes, values, units, entities, relationships, rankings, filters, and comparisons. The models produced from the statements in the computed outline are organized into a graph based on how the ideas are related to each other. The shape of the graph provides a method for the NLG system to handle phrase muting, clause placement, anaphora, and connectives.

[0098] For example, the statement for AttributeOfEntity (‘Sales’, EntityByID (‘1234’, Salesperson1234), 15000) is converted into a model graph where the root is an EntityModel representing the Salesperson1234. The EntityModel has a dependent AttributeModel representing the Sales attribute since Sales is an attribute of that entity. The attribute Sales has a value of 15000 so a ValueModel representing 15000 is added as a dependent to the AttributeModel. Finally, the ValueModel has a UnitModel representing the type of value. In this case it is ‘dollars’. This model graph now provides the structure needed for the NLG system to construct a sentence for this statement. This was a simple example. The more complicated the statement, the more complicated the model graph will be. The system can also combine multiple statements into a single big model graph assuming they are related somehow, for example each of them are about the same entity. This then allows the system to then express multiple sets of ideas in a single sentence. If the model graph is too big, i.e. there are too many ideas to express in one sentence, it is split up into reasonably sized subgraphs that make up individual sentences.

[0099] After a model graph has been generated for each node, adjacent nodes are compared with each other to mute redundant facts. This can be referred to as Model Muting, as shown by FIG. **8**B. Model Muting reduces redundant information from being expressed across sentences. Since the working example has only a single goal, there is only one node involved, and there will be nothing to mute in this phase with respect to the example. Say though, the goal also had a timeframe associated with it so instead it was “Present the sales in the month of the Sales Person” and an adjacent goal was “Present the sales in the month of the top ranking Sales Person by sales”. Without muting these goals would express as, “In August of 1993, Joe had sales of $15000. In August of 1993, Bob, the best seller, had sales of $430000”. The timeframe “In August of 1993” is redundant between these two sentences and will be dropped in the second sentence resulting in language of “In August of 1993, Joe had sales of $15000. Bob, the best seller, had sales of $430000”.

[0100] Next, sentences are generated based on each model graph during Sentence Generation as shown by FIG. **8**C. The base of the sentence is generated first. It is the core subject/verb/object constituents of a sentence. Initially this will not have expressed all of the models in the graph (those will be added later as clauses). Not all models in the graph can generate base sentences, but multiple models can add to the set of possible sentences for a node. Sentences almost always come from preferences set by the user in the ontology **320** through things like attribute expressions, rank expressions, and/or relationship expressions. The sentences generated in this phase will be built upon, and later one of these sentences will be picked to be used in the narrative story.

[0101] Continuing with the working example, only the Attribute model can generate sentences for this model graph. It will generate them based on the attribute expressions configured by the user for “sales”. Let's suppose the user configured three options: “the salesperson had sales of $100”, “the salesperson sells $100”, and “the salesperson's sales are $100”. The Attribute model would generate three sentences, one for each of these options.

[0102] After the base sentences have been generated, the models not expressed in that base sentence are then be expressed as clauses on the sentence. This can be referred to as Clause

Placement (see FIG. **8**D). Depending on where the unexpressed models are in the model graph, they will be placed as phrases on the sentence attached to the noun representing the model in the graph they are dependents of. This is done for each sentence from the list of sentences produced by the sentence generation phase. Clauses are generated similarly to how sentences were generated in the previous phase based on the user's expression preferences within the ontology.

[0103] In our example, there are no extra models that need to be added as clauses. However, to illustrate how the clause placement phase would work, let's say that the goal was actually “Present the sales of the salesperson working in the city.” A sentence from the Relationship model would be “Sally sells in Chicago.” This leaves the Attribute/Value/Unit models still needing to be expressed. The Attribute model can produce clauses for these. Based on the attribute expression configuration, it would generate clauses of “who has sales of $1000” or “who has sold $1000”. These would be added as a relative clause to “Sally” giving a complete sentence of “Sally, who has sales of $1000, sells in Chicago” (as one of the sentences among the several available permutations).

[0104] The next phase is Sentence Selection (see FIG. **8**E). At this point, complete sentences have been built, and the system needs to pick one for use in the narrative story. The Sentence Selection phase can take into consideration several factors when selecting sentences. For example, the selected sentence should (1) correctly convey the intent of the goal, (2) only express what is necessary, and (3) prefer patterns that generally sound better. With these criteria, the system will likely be still left with more than one valid sentence. At this point, the system can choose from the remaining sentences that provide the best variability of expression. In an example embodiment, with all factors being equal, the system can randomly select a sentence from among the qualifying sentences. In our example, based on the goal, all three sentences are equally valid, so the system will randomly choose one to include in the final story. At the conclusion of the Sentence Selection phase, a sentence will have been selected for each node in the outline.

[0105] At this point, the system seeks to improve fluidity by looking across the nodes in the outline. At this stage, referred to as Entity Referencing (see FIG. **8**F), nodes in the same section that repeat entities will be replaced with pronouns. The pronoun used will depend on the type of entity being replaced. If the base entity type is a Person and gender is available, the system will use gendered pronouns (e.g., he/she), otherwise it will use a non-gendered pronoun (e.g., they).

[0106] In our example, since there is only a single goal there would be no pronoun replacement. If instead there were two adjacent goals in the same section (e.g., “Present the sales of the salesperson” and “Present the title of the salesperson”, a pronoun would be used for the second sentence, resulting in the language “Sally had sales of $10000. She had the title VP of Sales.”

[0107] At this point, the sentences have been finalized. The next thing to do is ensure that the sentences are grammatically correct. This phase can be referred to as Realization (see FIG. **8**G). To perform realization, the system adds articles (definite—“the”—and indefinite—“a/an”), conjugates verbs, and adds punctuation. After realization, the system has the final language for use in the story.

[0108] Wrapping up the example, the realized sentence ends up being “Sally has sales of $10,000.” To get to that, the verb “has” was conjugated into present tense because the lack of a timeframe. The system can be configured to assume the timeframe is “now” in cases where no timeframe is specified in the communication goal statement. Also, the Realization phase inspects “sales” and determines that it was plural so an indefinite article was not needed. Finally, “Sally” is determined to be a name proper noun, which accordingly means that a definite article is not needed before “Sally”.

[0109] As a last step, which can be referred to as Document Generation (see FIG. **8**H), the system puts the realized language into a formatted document. Examples of suitable formats can include HTML, Microsoft Word documents, and JSON. The system returns the formatted document to the client.

Ontology Building:

[0110] FIGS. **9-13** depict example process flows that show how the ontology **320** can be built in response to user input, including user input during the process of composing communication goal statements. Appendix A included herewith is a user guide for an example narrative generation platform, where the user guide shows examples of GUI screens that demonstrate how the ontology **320** can be built in response to user input.

[0111] FIG. **9** depicts an example process flow for parameterizing a value in a communication goal statement, which relates to the attribute objects in the ontology **320**. It should be understood that the order of many of the steps in this process flow could be changed if desired by a practitioner. At step **900**, the processor determines in response to user input whether a new attribute should be created for the value to be parameterized or whether an existing attribute should be used. Appendix A depicts example GUI screens that can assist the user as part of this process (see, e.g., FIG. **159** et seq.). If an existing attribute is to be used, the system can access the ontology **320** to provide the user with a list of attributes available for selection by the user. The user can select an existing attribute from this list (step **918**). The system can also use string matching technology to match any characters entered by a user through the GUI to existing attributes in the ontology **320**. Upon detecting a match or partial match, the system can then suggest an existing attribute for selection.

[0112] If a new attribute is to be created for the value, the process flow proceeds to step **902**. At step **902**, the process flow makes a decision as to whether the new attribute should be a direct attribute or a computed value attribute.

[0113] If a direct attribute is to be created, the process flow proceeds to step **904**. At step **904**, the processor defines a label for the attribute in response to user input. This label can serve as the name for the attribute (e.g., "sales"-see FIG. **54**). Next, at step **906**, the processor defines a base type for the attribute in response to use input. Examples of base types for attributes can include currency, date, decimal, integer, percentage, and string. FIG. **55** shows an example GUI screen through which a user can set the type for the subject attribute.

[0114] Next, at step **908**, the processor defines the expression(s) that are to be associated with the subject attribute. Through specification of one or more expressions for the subject attribute, the user can provide the system with a number of options for expressing the attribute in words when rendering a narrative story.

[0115] At step **910**, the processor selects the entity type for the subject attribute in response to user input. FIGS. **56-61** show example GUI screens for step **910**. Step **910** is further elaborated upon with reference to FIG. **11** discussed below.

[0116] If step **902** results in a determination that a computed value attribute is to be created, the process flow proceeds to step **912** from step **902**. At step **912**, the system presents the user with a choice of making the computed value attribute a function or an aggregation (step **912**). If a function is selected at step **912**, the process flow proceeds to step **914** where the processor sets the computed value attribute according to the user-selected function. If an aggregation is selected at step **912**, the process flow proceeds to step **916** where the processor sets the computed value attribute according to the user-selected aggregation. Examples of available aggregations can include count, max, mean, median, min, range, and total. These aggregations can be associated with corresponding parameterized computational logic (see FIG. **3**D) that is programmed to compute the desired aggregation. An example of an available function is a contribution function, which evaluates how much a component contributes to an aggregate. However, it should be understood that other functions can be available through the system. For example, additional functions could include a multiplication, a division, a subtraction, standard deviation, a first derivative, and a second derivative. FIGS. **166-167**, described in greater detail below in Appendix A, illustrate some example GUI screens through which a user can define computed value attributes.

[0117] After the attribute has been defined via the process flow of FIG. **9**, the ontology **320** can be updated by adding the details for attribute **330** to ontology **320**.

[0118] It should be understood that additional operations can be included in the attribute definition process flow if desired by a practitioner. For example, if a practitioner wishes to attach timeframe details to attributes, a timeframe definition process flow can be added to the FIG. **9** process flow.

[0119] FIG. **10** depicts an example process flow for parameterizing a characterization object in a communication goal statement and ontology. Characterizations **332** are editorial judgments based on defined qualifications that determine the language used when certain conditions are met. Through a characterization **332**, a user is able to associate descriptive language with an entity type based on the nature of one or more attributes of that entity type. At step **1000**, the processor selects the entity type to be characterized in response to user input. FIG. **11** provides an example process flow that elaborates on how the entity type can be defined.

[0120] At step **1002**, the system determines whether the user wants to create a new characterization or select an existing characterization. This step can be performed in a manner similarly to step **900** in FIG. **9**, but for characterizations rather than attributes. If an existing characterization is desired, the system can make a selection of an existing characterization in response to user input at step **1012**. However, if a new characterization is desired, the process flow proceeds to step **1004**.

[0121] At step **1004**, the user selects the attribute(s) for use in the characterization. If the attribute needs to be defined, the process flow of FIG. **9** can be followed. For example, if the characterization **332** is meant to characterize the performance of a salesperson in terms of sales by the salesperson, step **1004** can result in the user selecting the attribute “sales” as the attribute by which the characterization will be determined.

[0122] At step **1006**, the user sets the qualification(s) by which to evaluate the characterization. For example, these qualifications can be a series of thresholds by which the values of the sales attribute are judged (e.g., the characterization changes based on whether the sales amount are above or below a threshold of $10,000). Multiple thresholds can be defined for a characterization, which would then yield more than two potential outcomes of a characterization (e.g., three or more tiers of characterization outcomes). Also, the qualifications need not be defined in terms of fixed thresholds. The thresholds can also be flexibly defined in terms of direct attributes and/or computed value attributes (for example, a salesperson can be characterized as a satisfactory salesperson if the sales attribute for the subject salesperson has a value that exceeds the value of the benchmark attribute for the subject salesperson; as another example, a salesperson can be characterized as an above-average salesperson if the sales attribute for the subject salesperson has a value that exceeds the average value of the sales attributes for the all of the salespeople within a company). As part of defining the qualifications, step **1006** can also involve the user specifying the operators by which to judge qualifications. Examples of operators may include “greater than”, “less than”, “greater than or equal to”, “equals”, etc.

[0123] At step **1008**, the user sets the expression(s) for the subject characterization. These expressions can then be used by the NLG process when articulating the subject characterization in a narrative story. For example, in a characterization relating to the performance of a salesperson in terms of sales, expressions such as “star performer”, “outperformed”, “high performer” etc. can be used in situations where the sales exceeded the highest threshold, while expressions such as “laggard”, “poor performer”, “struggled”, etc. can be used in situations where the sales were below the lowest threshold.

[0124] FIGS. **72-75**, **141-156**, and **199-204** depict example GUIs through which a user can provide inputs for the process flow of FIG. **10**. Upon the completion of the FIG. **10** process flow, the system can update the ontology **320** to add the details for the defined characterization **332**. It should be understood that additional operations can be included in the characterization definition process flow if desired by a practitioner. For example, if a practitioner wishes to attach timeframe details to characterization, a timeframe definition process flow can be added to the FIG. **10** process flow.

[0125] FIG. **11** depicts an example process flow for parameterizing an entity type in a communication goal statement and ontology. Entity types are how the system knows what to talk about with respect to a communication goal statement. An entity type is a primary object in the ontology which has particular attributes (e.g., a department (entity type) has expenses (attribute). An entity is a specific instance of an entity type, with data-driven values for each attribute (e.g., John Smith is a specific instance of a salesperson entity type, and this entity has a specific data value for the sales attribute of a salesperson entity type). Ontology **320** may include more than one entity type.

[0126] At step **1100**, the processor decides, in response to user input, whether to create a new entity type or select an existing entity type. This step can be performed while a user is composing a communication goal statement. If step **1100** results in a determination that an existing entity type is to be used, the process flow can proceed to step **1150** where an existing entity type is selected.

[0127] If step **1100** results in a determination that a new entity type is to be created, the process flow proceeds to step **1102**. At step **1102**, the user provides a label for the entity type. This label can be used as the entity type's name (e.g., a “salesperson” entity type). Next, at step **1104**, the user sets a base type for the subject entity type. Examples of available base types to choose from can include person, place, thing, and event. However, it should be understood that more, fewer, and/or different base types can be used. The specified base type can be used by the AI logic to inform decision-making about the types of pronouns that can be used to express the subject entity type, among other expressive qualities for the entity type.

[0128] At step **1106**, the user sets one or more expressions in relation to the subject entity type. These expressions provide the NLG process with a variety of options for expressing the entity type in a story.

[0129] The FIG. **11** process flow can also include options for attaching a number of additional features to entity types.

[0130] For example, a relationship can be added to the subject entity type at steps **1108-1116**. At step **1110**, the user identifies the entity type to which the subject entity type is to be related. If the relating entity type does not exist, the process flow of FIG. **11** can be recursively invoked to create the relating entity type. An example of a relating entity type might be a “company” entity type with respect to a subject entity type of “salesperson”. Steps **1112-1116** operate to define the nature of the relationship between the subject entity type and the relating entity type. At step **1112**, the process flow determines whether the user wants to create a new relationship or select an existing relationship. If create new is selected at step **1112**, the process flow proceeds to step **1114** where the user provides an expression for the new relationship (e.g., the relating expression can be “employed by” to relate the subject entity type of “salesperson” to the relating entity type of “company” (thus, the “salesperson” is “employed by” the “company”). Multiple expressions may be provided at step **1114** to provide variability during story rendering. For example, the expressions “works for”, “is a member of”, “belongs to” might be used as alternative expressions for the relationship between the “salesperson” entity type and the “company” entity type. If select existing is selected at step **1112**, the process flow proceeds to step **1116** where a user can be presents with a list of existing relationship expressions known to the system or within the ontology. The user can then select one or more of these expressions to define the nature of the relationship between the subject entity type and the relating entity type.

[0131] Another example of a feature that can be added to an entity type is a rank. Steps **1120-1124** describe how a rank can be attached to an entity type. The rank feature provides the AI with a mechanism for notionally identifying entities to be discussed in a narrative story even if the user does not know in advance which specific entities are to be discussed. For example, a user may want the system to generate a story about the 3 top ranked salespeople in terms of sales, but does not know a priori who these salespeople are. The rank feature attached to the salesperson entity type allows for a user to easily compose a communication goal statement that can be used by the AI to generate an appropriate narrative story. At step **1122**, the user sets the attribute by which the subject entity type is to be ranked. For example, if salespeople are to be ranked by sales, the user can specify the sales attribute at step **1122**. The FIG. **9** process flow can be followed to specify the subject attribute for ranking. At step **1124**, the user sets a rank slice for the rank feature. The rank slice defines a depth for the rank feature with respect to the subject entity type. If the rank slice is set to 1, only the top ranked entity would be applicable. If the rank slice is set to n, the n highest rank entities would be returned.

[0132] Another example of a feature that can be added to an entity type is a qualification. Steps **1130-1134** describe how a qualification can be attached to an entity type. Similarly to the rank feature, the qualification feature provides the AI with a mechanism for notionally identifying entities to be discussed in a narrative story even if the user does not know in advance which specific entities are to be discussed. For example, a user may want the system to generate a story about the salespeople who have 10 years of more of experience or who have been characterized as star performers in terms of sales, but does not know a priori who these salespeople are. The qualification feature attached to the salesperson entity type allows for a user to easily compose a communication goal statement that can be used by the AI to generate an appropriate narrative story. At step **1132**, the user sets the attribute **330** and/or characterization **332** that will be used to filter/qualify the subject entity type. For example, if the user wants the story to focus on salespeople with at least 10 years of experience, the user can specify a “years worked” or “start date” attribute at step **1132**. The FIG. **9** process flow can be followed to specify the subject attribute for qualification. If a user wants to specify a characterization at step **1132**, the FIG. **10** process flow can be followed in order to specify a characterization of qualification. At step **1134**, the user defines condition(s) for the qualification. For example, if a “years worked” attribute is set as the qualification and the user wants to qualify salespeople based on 10 years of experience, the user can define the condition on the attribute as 10 years.

[0133] FIGS. **116-156** depict example GUIs through which a user can provide inputs for the process flow of FIG. **11**. Upon the completion of the FIG. **11** process flow, the system can update the ontology **320** to add the details for the defined entity type **322**. It should be understood that additional operations can be included in the entity type definition process flow if desired by a practitioner. For example, if a practitioner wishes to attach timeframe details to characterization, a timeframe definition process flow can be added to the FIG. **11** process flow. As another example, the FIG. **11** process flow can include branching options for adding an attribute to an entity type directly from the FIG. **11** process flow if desired. Similarly, the FIG. **11** process flow can also include branching options for adding a characterization to an entity type directly from the FIG. **11** process flow if desired.

[0134] FIG. **12** depicts an example process flow for parameterizing a timeframe in a communication goal statement and ontology. A timeframe is a unit of time used as a parameter to constrain the values included in the expression of a communication goal statement or narrative story. Ontology **320** may include more than one timeframe.

[0135] At step **1200**, the processor decides, in response to user input, whether to create a new timeframe or select an existing timeframe. This step can be performed while a user is composing a communication goal statement. If step **1200** results in a determination that an existing timeframe is to be used, the process flow can proceed to step **1212** where an existing timeframe is selected.

[0136] If step **1200** results in a determination that a new timeframe is to be created, the process flow proceeds to step **1202**. At step **1202**, the system determines whether the user wants to create a new timeframe type or select from among existing timeframe types. Examples of timeframe types include years, months, days, hours, etc.

[0137] If a new timeframe type is desired, the process flow proceeds to step **1204** where the user defines the timeframe type and step **1206** where the user sets the expression(s) for the timeframe type. The expression(s) provide the NLG process with a variety of options for expressing the timeframe in a story.

[0138] If an existing timeframe type is desired, the process flow proceeds to step **1208** where the user makes a selection from among existing timeframe types and step **1210** where the user defines a designation for the selected timeframe type. Through this designation, the user can define qualifications via a "when" statement or the like that defines time-based conditions (e.g., "the month of the year when the sales of the store were highest").

[0139] FIGS. **62-64**, **87-88**, **96**, **102**, **162-165**, **187**, and **196-198** depict example GUIs through which a user can provide inputs for the process flow of FIG. **12**. Upon the completion of the FIG. **12** process flow, the system can update the ontology **320** to add the details for the defined timeframe **344**.

[0140] FIG. **13** depicts an example process flow for parameterizing a timeframe interval for use with a timeframe. The timeframe interval defines how the system should consider intervals of time within a timeframe (e.g., days of the month, weeks of the month, months of the year, quarters of the year, hours of the day, etc.). At step **1300**, the processor decides, in response to user input, whether to create a new timeframe interval or select an existing timeframe interval. If step **1300** results in a determination that an existing timeframe interval is to be used, the process flow can proceed to step **1306** where an existing timeframe interval is selected. If step **1300** results in a determination that a new timeframe interval is to be created, the process flow proceeds to step **1302**. At step **1302**, the user defines the timeframe interval, and at step **1204** the user sets one or more expression(s) for the timeframe interval. The expression(s) provide the NLG process with a variety of options for expressing the timeframe interval in a story. Upon the completion of the FIG. **13** process flow, the system can update the ontology **320** to add the details for the defined timeframe interval.

[0141] As explained above, the ontology **320** defined via the process flows of FIGS. **9-13** can be leveraged by the AI in coordination with the composed communication goal statements to not only determine the content to be expressed in the narrative story but also to determine how that content should be expressed in the narrative story.

Subgoals within Communication Goal Statements:

[0142] The communication goal statements may be interpreted by the system to include a plurality of subgoals or related goals. Thus, in order for a narrative story to satisfy the communication goal associated with a communication goal statement, it may be desirable to the narrative story to first satisfy one or more subgoals related to the communication goal of the communication goal statement. An example of this is shown by FIGS. **14**A-D. As shown by FIG. **14**A, a communication goal statement **1400** may be associated with a parent or base communication goal. The interpreter **506** may be configured to interpret communication goal statement **1400** as being comprised of two or more communication goal statements **1402** and **1404**, where these communication goal statements **1402** and **1404** are associated with subgoals relating to the parent/base goal. When the AI **502** seeks to determine the content for inclusion in the story, the interpreter **506** will process the communication goal statements **1402** and **1404** when generating the computed outline.

[0143] FIG. **14**B shows an example of this. In this example, the base communication goal statement corresponding to the parent/base goal is "Compare Value 1 to Value 2" (see base communication goal statement **406**). This base communication goal statement **406** can be comprised of a series of three base communication goal statements, each relating to subgoals of the parent/base goal. In this example, these three base communication goal statements are: (1) "Present Value 1" **402**.sub.1, (2) "Present Value 2" **402**.sub.2, and (3) "Characterize the Difference Between Value 1 and Value 2" **404**. Thus, for the narrative story to accomplish the overall parent/base goal of comparing Value 1 to Value 2, it will be helpful for the narrative story to first present Values 1 and 2 and then provide a characterization of the difference between Values 1 and 2.

[0144] During the composition process, a user may parameterize the base communication goal statement **406** of FIG. **14**B as shown by FIG. **14**C. As shown by FIG. **14**C, the parameterized communication goal statement **406*b*** can read "Compare the Sales of the Salesperson during the Timeframe to the Benchmark of the Salesperson", where Value 1 is the "Sales of the Salesperson during the Timeframe" and Value 2 is the "Benchmark of the Salesperson". The interpreter **506** can be configured to interpret parameterized communication goal statement **406*b*** for the purposes of story generation as the following three parameterized communication goal statements: (1) “Present the Sales of the Salesperson during the Timeframe” **402**.sub.1b, (2) “Present the Benchmark of the Salesperson” **402**.sub.2b, and (3) “Characterize the Difference Between the Sales of the Salesperson during the Timeframe and the Benchmark of the Salesperson” **404***b*. The system can then interact with ontology **320** to generate a narrative story as shown by FIG. **14**D from these three parameterized communication goal statements. As can be seen by FIG. **14**D, the NLG process created the first sentence of the narrative story in a compound form to satisfy the subgoals associated with the first two parameterized communication goal statements **402**.sub.1*b* and **402**.sub.2*b*. The final sentence of the narrative story satisfies the subgoal associated with the third parameterized communication goal statement **404***b*. Overall, the narrative story satisfies the parent/base goal associated with parameterized communication goal statement **406***b*.

[0145] During the process of composing communication goal statements for use in the narrative generation process, the system can provide GUI screens to a user that allows the user to expand a communication goal statement to show communication goal statements associated with subgoals. Furthermore, the GUI can be configured to respond to user input to selectively opt in and opt out of which subgoals are to be included in the narrative generation process for a section of the story outline. Thus, if a user wants the story to include a headline or a title that is drawn from the “Compare” communication goal statement, a user can use a GUI to expand the “Compare” communication goal statement into statements for its constituent subgoals. For the headline/title, a user can choose to selectively opt out of the first two “Present” statements but retain the “Characterize” statement so that the headline/title is focused on a desired main point. Then, in the body of the narrative story, the user can selectively retain all of the constituent subgoals for the “Compare” statement so that the body of the narrative story provides the context for the comparison. FIGS. **70-71** and **210** depict example GUIs through which a user can expand a communication goal statement to view its related subgoals and selectively choose which of the subgoals will be used during the narrative generation process.

Example Embodiments for a Conditional Outcome Framework to Determine Narrative Content:

[0146] In another example embodiment, the system can employ a conditional outcome framework to support narrative generation. For example, AI **502** can employ a conditional outcome framework to determine content for inclusion in a narrative. FIG. **15**A illustrates a simplified example where a conditional outcome data structure **1502** is linked with one or more idea data structures **1504**, where each idea data structure **1504** represents an idea that is to be expressed in a narrative. The conditional outcome structure **1502** can comprise (1) a name corresponding to the conditional outcome, (2) one or more conditions that define when the conditional outcome is defined as true, and (3) one or more links to one or more content or idea structures **1502**/**1504**. Thus, the conditional outcome data structure provides a mechanism for analyzing data to intelligently determine what ideas should be expressed in a narrative about that data. This can serve as a powerful building block for constructing the AI **502** in a manner so that the content expressed in a narrative will intelligently respond to the underlying data being considered.

[0147] FIG. **15**B depicts an example that shows how the conditional outcome framework can be used in combination with a communication goal statement to intelligently adapt narratives to their underlying data in a manner that satisfies a desired communication goal. In FIG. **15**B, narrative analytics **510** employ a conditional outcome framework **1500**. As explained in connection with FIG. **5**, the narrative analytics **510** can be associated with a communication goal statement **390**. Thus, as the system processes a communication goal statement **390**, an appropriate set of narrative analytics **510** tailored toward satisfying that communication goal statement can be selected. The conditional outcome framework **1500** can include one or more outcome data structures **1502** linked with one or more idea data structure **1504** as discussed above in connection with FIG. **15**A. Furthermore, any of the outcome data structures **1502** and/or idea data structures **1504** can be associated with supporting analytics **1506**. The supporting analytics provide logic that can be used by the system to compute information used for navigating the conditional outcome framework **1500** and identifying ideas during execution at **526** (see FIG. **5**).

[0148] It should be understood that the outcome data structures **1502** can be tied together in numerous arrangements to define branching logic for the conditional outcome framework **1500**. For example, there can be multiple layers of outcome data structures **1502** (each with associated conditions) to provide branching operations at multiple levels. Such branching structures allow for the conditional outcome framework **1500** to accommodate highly complex and intelligent decision-making as to what ideas should be expressed in a narrative in view of the nature of the data under consideration. Moreover, the outcome data structures **1502**, idea data structures **1504**, and supporting analytics **1506** can be parameterized to allow their re-use in a wide variety of contexts.

[0149] It should also be understood that the same idea data structure **1504** might be linked to multiple different outcome data structures **1502**. Furthermore, a given outcome data structure **1502** might be linked to multiple idea data structures **1504**. Examples of such arrangements are discussed below with reference to FIG. **16** et seq.

Example Embodiments for "Analyze" Communication Goal Statements:

[0150] As mentioned above, an operator such as "Analyze" can be used to identify a communication goal statement corresponding to an analysis communication goal. An example of a base communication goal statement for an analysis communication goal that could be supported by the system is "Analyze Entity Group by Attribute", where "Entity Group" serves as a parameter for a group of entities in the ontology **320** and "Attribute" serves as a parameter for an attribute of the specified entity group in the ontology **320**. Such a base communication goal statement could be parameterized into a communication goal statement as "Analyze the Salespeople by Sales", where the Entity Group is specified as "Salespeople" (which can be a group of entities in the ontology **320** that have the entity type of "Salesperson"), and where the Attribute is specified as "Sales" (which can be an attribute of a "Salesperson" in the ontology **320**). However, it should be understood that such a base communication goal statement could be parameterized in any of a number of different ways. Further still, it should be understood that different base communication goal statements could be used to satisfy other analysis-related communication goals, some examples of which are discussed below.

[0151] The system can link a base communication goal statement of "Analyze Entity Group by Attribute" with narrative analytics **510** that are linked to a story structure that aims to provide the reader with an understanding of the distribution of a particular value across a group of entities. Accomplishing this may involve expressing a variety of quantitative ideas (the number of entities in the group, the average value within a group, the median value within a group, the entities with the highest and lowest values, etc.) and more qualitative ideas (the values are distributed normally, the values are distributed exponentially, the values demonstrate a "long-tail" distribution, one entity in particular had a much higher value than the other entities, etc.). Accordingly, if desired by a practitioner, the system can directly map such a communication goal statement to parameterized narrative analytics and a parameterized story configuration that will express these concepts. However, the use of a conditional outcome framework **1500** by the relevant narrative analytics can provide additional flexibility where the resulting narrative story structure will adapt as a function of not only the specified communication goal but also as a function of the underlying data.

[0152] FIG. **16** discloses an example embodiment for a conditional outcome framework that can be used by the narrative analytics **510** associated with a communication goal statement **390** for "Analyze Entity Group by Attribute". In this example, the conditional outcome framework can employ multiple levels or layers of outcomes **1502**. For example, a first layer of outcomes **1502** can correspond to different conditional outcomes that characterize the size of the group specified in the communication goal statement **390**. The second layer of outcomes **1502** can correspond to different conditional outcomes that characterize the distribution of group members within the group based on the attribute specified by the communication goal statement **390**. The first layer conditional outcomes **1502** can include a “tiny group” outcome **1502**, a “decent sized group” outcome **1502**, and a “large group” outcome **1502**. Each of these different conditional outcomes **1502** can be tied to the conditions that are evaluated by the system to assess whether that conditional outcome **1502** fits the underlying data.

[0153] To drive the assessments regarding group size, the supporting analytics **1506** for the conditional outcome framework can include group size characterization analytics **1600** for the various group size outcomes **1502**. For example, the “tiny group” outcome **1502** can be associated with parameterized logic that determines whether the number of members of the group specified by the communication goal statement **390** is less than or equal to 1 (it should be understood that other thresholds could be used to define the boundary conditions for a “tiny group”). If so, the “tiny group” outcome **1502** would evaluate as true. As another example, the “decent sized group” outcome **1502** can be associated with parameterized logic that determines whether the number of members of the group specified by the communication goal statement **390** is between 2 and 50 (it should be understood that other thresholds could be used to define the boundary conditions for a “decent sized group”). If so, the “decent sized group” outcome **1502** would evaluate as true. As another example, the “large group” outcome **1502** can be associated with parameterized logic that determines whether the number of members of the group specified by the communication goal statement **390** exceeds 50 (it should be understood that other thresholds could be used to define the boundary conditions for a “large group”). If so, the “large group” outcome **1502** would evaluate as true.

[0154] To drive the assessments regarding distribution within the group, the supporting analytics **1506** for the conditional outcome framework can include group distribution characterization analytics **1602** for the various group distribution outcomes **1502**. In this example, the system seeks to characterize (1) a “tiny group” as being an empty group (see the “empty” outcome **1502**) or a single member group (see the “just one” outcome **1502**), (2) a “decent sized group” as being a typical distribution (see “typical distribution” outcome **1502**), a distribution that is clumpy at the top (see “clump at top” outcome **1502**), or a flat distribution (see the “flat distribution” outcome **1502**), and (3) a “large group” as being a normal distribution (see “normal distribution” outcome **1502**) or a long-tail distribution (see the “long-tail distribution” outcome **1502**). Each of these second level outcomes **1502** can be associated with parameterized analytics **1602** that specify the computations used for characterizing the nature of the distributions within the group. For example, the “clump at top” outcome **1502** can be associated with parameterized analytics **1602** that are configured to sort entities by a particular value, group entities with similar values, and then determine if the highest ranked entities constitute a subgroup of similar values. Any thresholds or parameters used in determining such subgroups may be built into the system, specified directly by users, or tuned automatically by the system. As another example, the “long-tail distribution” outcome **1502** can be associated with parameterized analytics **1602** that are configured to perform distribution analysis and then determine if a significant proportion of the entities contributed values well below the mean contribution. Again, any thresholds or parameters used could be built into the system, specified directly by users, or tuned automatically by the system.

[0155] In FIG. **16**, each second layer/level outcome **1502** is linked to one or more idea data structures **1504**. Thus, the resolution of which ideas should be expressed in a given narrative that is generated to satisfy the communication goal statement **390** will depend on which outcomes **1502** were deemed true in view of the underlying data. The relationships between ideas for expression in a narrative to the nature of the underlying data in this example can be seen in the table below: TABLE-US-00001 Outcome of Ideas to be Expressed in the Characterizing the Narrative About the Underlying Data Underlying Data Tiny Group Narrative should express the following idea: (Empty Set) A count of the group members Tiny Group Narrative should express the following idea: (Single Member) A count of the group members Decent Sized Narrative should express the following ideas: Group (Typical A count of the group members Distribution) The total of the attribute values for the group The mean of the attribute values for the group The names and values of the top N group members as ranked according to the group members' associated attribute values. Decent Sized Narrative should express the following ideas: Group (Clump at A count of the group members Top Distribution) The total of the attribute values for the group The mean of the attribute values for the group A discussion of the clumpy nature of the distribution of members within the group with respect to the attribute values. The names and values of the group members in the top clump (as ranked according to the group members' associated attribute values). Decent Sized Narrative should express the following ideas: Group (Flat A count of the group members Distribution) The total of the attribute values for the group The mean of the attribute values for the group A discussion of the flat nature of the distribution of members within the group with respect to the attribute values. Large Group Narrative should express the following ideas: (Normal A count of the group members Distribution) The mean of the attribute values for the group The names and values of the group members in the top n percentile (as ranked according to the group members' associated attribute values). Large Group Narrative should express the following ideas: (Long Tail A count of the group members Distribution) The total of the attribute values for the group A discussion of the long tail nature of the distribution of members within the group with respect to the attribute values. The names and values of the group members in the top n percentile (as ranked according to the group members' associated attribute values).

[0156] Any ideas **1504** that are resolved based on the conditional outcome framework could then be inserted into the computed story outline **528** for use by AI **504** (together with their associated specifications in view of the underlying data) when rendering the desired narrative.

[0157] To the extent that any of the ideas **1504** need additional computed values in order to be expressed (where such values were not previously computed by analytics **1600** or **1602**), the supporting analytics **1506** can further include idea support analytics **1604**. For example, if the analytics **1600** and **1602** do not compute a mean value for the attribute values within the group, the idea support analytics **1604** can include parameterized logic that computes such a mean value for the underlying data.

[0158] Thus, it can be seen that the example conditional outcome framework for a communication goal statement can define a hierarchical relationship among linked outcomes and ideas together with associated supporting analytics to drive a determination as to which ideas should be expressed in a narrative about a data set, where the selection of ideas for expression in the narrative can vary as a function of the nature of the data set.

[0159] In example embodiments, the conditional outcome framework can be designed so that it does not need any input or configuration from a user other than what is used to compose the communication goal statement **390** (e.g., for the “Analyze Entity Group by Attribute” communication goal statement, the system would only need to know the specified entity group and the specified attribute). However, for other example embodiments, a practitioner might want to expose some of the parameters of the conditional outcome framework to users to allow further configurations or adjustments of the conditional outcome framework.

[0160] For example, a practitioner might want to implement the thresholds used within the conditional outcome framework as user-defined values. In the context of FIG. **16**, this could involve exposing the thresholds used for characterizing the size of the group to users so that a user can adjust the group size boundaries in a desired manner (e.g., in some contexts, a large group might have a minimum of **100** members, while in other contexts a large group might have a minimum of **1000** members). Similarly, the values for “n” used by the conditional outcome framework of FIG. **16** (e.g., the top “n” group members or the “nth percentile”) could be exposed to users to allow adjustments of the value used for n.

[0161] As another example, a practitioner might want to provide users with a capability to enable/disable the links between outcomes **1502** and ideas **1504** in a conditional outcome framework. For example, a GUI could present a user with lists of all of the outcomes **1502** and ideas **1504** that can be tied to a communication goal statement within a conditional outcome framework. The user could then individually select which ideas **1504** are to be linked to which outcomes **1502**. If desired by a practitioner, that conditional outcome framework can include default linkages that are presented in the GUI, and the user could make adjustments from there. FIG. **17**A shows an example where a user has adjusted the conditional outcome framework to add a linkage **1700** between the "present the mean" idea **1504** and the "long tail distribution" outcome **1502**. FIG. **17**B shows an example where a user has removed the linkages **1702** that had previously existed between the "present the mean" idea **1504** and the "typical distribution", "clump at top", "flat distribution", and "normal distribution" outcomes **1502**.

[0162] FIG. **18**A shows an example of a narrative **1802** that can be generated using the conditional outcome framework of FIG. **16** as applied to a communication goal statement **1800** of "Analyze the salespeople by bookings" with respect to a data set that includes various salespeople and their associated bookings (e.g., the dollar values of their bookings). In this example, the narrative **1802** would be generated after an analysis of the data set arrived at a determination that the outcomes **1804** were true (the salespeople group was "decently sized" and has a "typical distribution" of salespeople with respect to their bookings). As can be seen in FIG. **18**A, the narrative text **1802** expresses the following ideas **1806** that are tied to the outcomes **1804**: (1) a count of the number of salespeople in the group, (2) the total amount of bookings for the salespeople in the group, (3) the mean value of bookings for the salespeople in the group, and (4) the names of the top 3 salespeople in the group (by the booking values) and the booking values for each of the top 3.

[0163] FIG. **18**B shows an example of a narrative **1812** that can be generated using the conditional outcome framework of FIG. **16** as applied to a communication goal statement **1810** of "Analyze the citizens by their salary" with respect to a data set that includes various citizens and their associated salaries. In this example, the narrative **1812** would be generated after an analysis of the data set arrived at a determination that the outcomes **1814** were true (the citizens group was a "large group" and has a "normal distribution" of citizens with respect to their salaries). As can be seen in FIG. **18**B, the narrative text **1812** expresses the following ideas **1816** that are tied to the outcomes **1814**: (1) a count of the number of citizens in the group, (2) the mean value of the salaries for the citizens in the group, and (3) the average salary of the top decile of citizens (with respect to their salaries).

[0164] FIGS. **18**A and **18**B thus show how the same parameterized conditional outcome framework can be used to generate narrative stories across different content verticals (e.g., a story about salespeople and their bookings as in FIG. **18**A versus a story about citizens and their salaries as in FIG. **18**B), which demonstrates how the parameterized conditional outcome framework provides an effective technical solution to the technical problem of horizontal scalability in the NLG arts.

[0165] It should be understood that the system can also be designed to support other "analyze" communication goals. For example, another base communication goal statement that can be used by the system can be "Analyze Entity Group by Attribute **1** and Attribute **2**". Such a multi-attribute analysis goal can trigger the performance of tradeoff analysis as between the two attributes (and the expression of ideas that result from this analysis). For example, this goal may trigger analysis that results in quantitative ideas like the average values for Attribute **1**, the average values for Attribute **2**, the entity with the largest value for Attribute **1**, etc. Assuming the system has an understanding of the relationship between Attribute **1** and Attribute **2** (for instance that "Attribute **1** is a driver of Attribute **2**" or that higher values for Attribute **1** represent a positive outcome while higher values for Attribute **2** represent a negative outcome), the goal may also result in more qualitative ideas that capture intuitive understandings like "Entities that score have high values for Attribute **1** also have high values for Attribute **2**", "The entity with the highest value for Attribute **1** actually has a really low value for Attribute **2**", or "There's no correlation between values for Attribute **1** and Attribute **2** in the group". Accordingly, it should be understood that it may be desirable for the narratives produced in response to the "Analyze Entity Group by Attribute **1** and Attribute **2**" communication goal statement to express different ideas than the narratives produced in response to the "Analyze

Entity Group by Attribute" communication goal statement.

[0166] FIGS. **19**A and B disclose an example embodiment for a conditional outcome framework that can be used by the narrative analytics **510** associated with a communication goal statement **390** for "Analyze Entity Group by Attribute **1** and Attribute **2**". In these examples, the outcomes can be associated with group size characterization analytics **1600** and group distribution characterization analytics **1602** as discussed above in connection with FIG. **16**. However, these outcomes can be linked to different ideas (and associated idea support analytics **1604**) as indicated by FIGS. **19**A and B. For example, the ideas of FIGS. **19**A and B can include totals, means, and names/values for the top n with respect to each attribute of the communication goal statement **390**. The ideas can also express whether the distributions of salespeople with respect to the two attributes are similar to each other or different than each other.

[0167] FIG. **19**A shows an example of a narrative **1902** that can be generated using the conditional outcome framework shown by the upper portions of FIG. **19**A-B as applied to a communication goal statement **1900** of "Analyze the salespeople by bookings and count of deals" with respect to a data set that includes various salespeople and their associated bookings (e.g., the dollar values of their bookings) and counts of their sales deals. In this example, the narrative **1902** would be generated after an analysis of the data set arrived at a determination that the outcomes **1904** were true (the salespeople group was a "tiny group" with only a single member). As can be seen in FIG. **19**A, the narrative text **1902** expresses the following ideas **1906** that are tied to the outcomes **1904**: (1) a count of the number of salespeople in the group, (2) the names of the top n salespeople in the group (by the first attribute, bookings value) and the booking values for each of the top n salespeople (which in this example is a single person's bookings), and (3) the names of the top n salespeople in the group (by the second attribute, deal count) and the count of deals for each of the top n salespeople (which in this example is a single person's deals).

[0168] FIG. **19**B shows an example of a narrative **1912** that can be generated using the conditional outcome framework shown by the upper portions of FIGS. **19**A-B as applied to the same communication goal statement **1900** shown by FIG. **19**A ("Analyze the salespeople by bookings and count of deals") but with respect to a different data set that includes various salespeople and their associated bookings (e.g., the dollar values of their bookings) and counts of their sales deals. In this example, the narrative **1912** would be generated after an analysis of the data set arrived at a determination that the outcomes **1914** were true (the salespeople group was a "decent sized group" and has similar distributions of values among the salespeople with respect to the two attributes, bookings and deal counts). As can be seen in FIG. **19**B, the narrative text **1912** expresses the following ideas **1916** that are tied to the outcomes **1914**: (1) a count of the number of salespeople in the group, (2) the total value of the first attribute (bookings) for the salespeople group, (3) the total value of the second attribute (deal counts) for the salespeople group, (4) the mean value of the first attribute (bookings) for the salespeople group, (5) the mean value of the second attribute (deal counts) for the salespeople group, (6) the names and attribute values for the top n of the salespeople group with respect to the first attribute (bookings), (7) the names and attribute values for the top n of the salespeople group with respect to the second attribute (deal counts), and (8) a statement that the distributions of salespeople with respect to the two attributes were similar to each other. FIGS. **19**A and B thus show how the same conditional outcome framework and same communication goal statement can produce dramatically different stories based on the content of the data set under consideration.

[0169] Another example of a base communication goal statement for an "analyze" communication goal that can be used by the system can be "Analyze Entity Group by a Change in Attribute (Over Time)". Such communication goal statement can trigger analysis that eventually results in quantitative ideas representing the total change in value, average change in value, the median change in value, which entity had the biggest change in values, the number of entities that had positive changes, etc. Such a goal might also produce more qualitative ideas that capture intuitive understandings such as "All members of the group had positive changes", "About half of the group had positive changes and about half had negative changes", or "The group as a whole had a positive change, but it was really a small group of entities that had large positive changes while the rest had smaller negative changes. A practitioner may desire that narratives produced from this communication goal statement express different ideas than those generated from the other "analyze" communication goals discussed above.

[0170] FIG. **20**A discloses an example embodiment for a conditional outcome framework that can be used by the narrative analytics **510** associated with a communication goal statement **390** for "Analyze Entity Group by a Change in Attribute (Over Time)". In this example, the framework includes attribute change analytics **2008** that computes the changes/deltas in the specified attribute values for each member of the entity group over the relevant time period. These deltas can then be used as the attribute values for the conditional outcome framework that can otherwise function as shown by FIG. **16**.

[0171] FIG. **20**A shows an example of a narrative **2002** that can be generated using the conditional outcome framework shown by the upper portion of FIG. **20**A as applied to a communication goal statement **2000** of "Analyze the salespeople by the change in their bookings" (where the relevant time frame can be either a default timeframe, system-determined time frame, or user-determined time frame, in this case corresponds to a time frame of Q1 to Q2) with respect to a data set that includes various salespeople and their associated bookings (e.g., the dollar values of their bookings) over time. In this example, the narrative **2002** would be generated after an analysis of the data set arrived at a determination that the outcomes **2004** were true (the salespeople group was a "decent sized group" with a typical distribution of attribute delta values for the salespeople). As can be seen in FIG. **20**A, the narrative text **2002** expresses the following ideas **2006** that are tied to the outcomes **2004**: (1) a count of the number of salespeople in the group, (2) the total number of salespeople in the group, (3) the mean value of changed bookings from Q1 to Q2 for the salespeople group, and (4) the names of the top n salespeople in the group (by their associated booking value deltas) and the booking value deltas for each of the top n salespeople.

[0172] FIG. **20**B discloses another example embodiment for a conditional outcome framework that can be used by the narrative analytics **510** associated with a communication goal statement **390** for "Analyze Entity Group by a Change in Attribute (Over Time)". In this example, the framework includes group size change characterization analytics **2010**, where these analytics **2010** are configured to analyzed the specified entity group to assess how its size changed over the relevant time period. In the example of FIG. **20**B, there are three outcomes associated with these analytics **2010**-*a* conclusion that the group size increased significantly, a conclusion that the group size stayed mostly consistent, and a conclusion that the group sized decreased significantly. To reach these outcomes, the analytics **2010** can tie each outcome to thresholds that are applied to computed changes in group size for the relevant time frame. For example, a group size change of +25% or more can be characterized as a significant increase, a group size change of −25% or more can be characterized as a significant decrease, and group sizes changes between these bounds can be characterized as consistent. Other outcomes within the conditional outcome framework can assess the nature of any change with respect to how the group members are ranked by the attribute over the relevant time frame. The analytics for these outcomes can also be parameterized to test whether their corresponding outcomes are applicable to the subject data. Furthermore, FIG. **20**B shows how the various ideas tied to the outcomes can include various informational items tied to the starting and ending times for the subject time frame, as well as ideas that express how certain group members rankings changed over the time frame.

[0173] FIG. **20**C shows an example of a narrative **2022** that can be generated using the conditional outcome framework shown by FIG. **20**B as applied to the communication goal statement **2000** of "Analyze the salespeople by the change in their bookings (over Q1 and Q2)" with respect to a data set that includes various salespeople and their associated bookings (e.g., the dollar values of their bookings) over time. In this example, the narrative **2022** would be generated after an analysis of the data set arrived at a determination that the outcomes **2024** were true (the size of the salespeople group increased significantly over Q1 to Q2, with the leaders among the salespeople with respect to bookings being largely unchanged over Q1 to Q2). As can be seen in FIG. **20**C, the narrative text **2022** expresses the following ideas **2026** that are tied to the outcomes **2024**: (1) an identification of the change in size for the salespeople group from Q1 to Q2, (2) a count of the members of the salespeople group at Q1, (3) a count of the members of the salespeople group at Q2, (4) the total amount of bookings for the salespeople group at Q1, (5) the total amount of bookings for the salespeople group at Q2, (6) the mean value of bookings for the salespeople group at Q2, and (7) the names and booking values for the top n salespeople at Q2 (in terms of bookings value).

[0174] FIG. **20**D shows an example of a narrative **2032** that can be generated using the conditional outcome framework shown by FIG. **20**B as applied to the same communication goal statement **2000** shown by FIG. **20**C ("Analyze the salespeople by the change in their bookings (over Q1 and Q2)") but with respect to a different data set that includes various salespeople and their associated bookings (e.g., the dollar values of their bookings) over time. In this example, the narrative **2032** would be generated after an analysis of the data set arrived at a determination that the outcomes **2034** were true (the size of the salespeople group decreased significantly over Q1 to Q2, with the salespeople who were leaders at Q1 with respect to bookings having been surpassed in Q2). As can be seen in FIG. **20**D, the narrative text **2032** expresses the following ideas **2036** that are tied to the outcomes **2034**: (1) an identification of the change in size for the salespeople group from Q1 to Q2, (2) a count of the members of the salespeople group at Q1, (3) a count of the members of the salespeople group at Q2, (4) the total amount of bookings for the salespeople group at Q1, (5) the total amount of bookings for the salespeople group at Q2, (6) the names and booking values for the top n salespeople at Q1 (in terms of bookings value), (7) the names and booking values for the top n salespeople at Q2 (in terms of bookings value), (8) the positions at Q2 of the salespeople who were in the top n at Q1, (9) the positions at Q1 of the sales people who were in the top n at Q2, and (10) a statement that notes the change in leadership for salespeople as between Q1 and Q2. FIGS. **20**C and **20**D thus show another example of how the same conditional outcome framework and same communication goal statement can produce dramatically different stories based on the content of the data set under consideration.

[0175] Yet another example of a base communication goal statement for an "analyze" communication goal that can be used by the system can be "Analyze Entity Group by Characterization". Such communication goal statement can trigger analysis that eventually results in quantitative ideas representing the count and percentage of entities with each characterization, the most common characterization, etc. Such a goal might also produce more qualitative ideas that capture intuitive understandings such as "There was a roughly even distribution of characterizations across the group", "Every entity in the group had the same characterization", "Almost all of the entities in the group had the same characterization", etc. A practitioner may desire that narratives produced from this communication goal statement express different ideas than those generated from the other "analyze" communication goals discussed above.

[0176] FIGS. **21**A and B disclose an example embodiment for a conditional outcome framework that can be used by the narrative analytics **510** associated with a communication goal statement **390** for "Analyze Entity Group by Characterization". In these examples, the outcomes can be associated with group size characterization analytics **1600** and group distribution characterization analytics **1602** as discussed above in connection with FIG. **16**. However, these outcomes can be linked to different ideas (and associated idea support analytics **1604**) as indicated by FIGS. **21**A and B. For example, the ideas of FIGS. **21**A and B can express concepts such as which characterizations are most common among members of the entity group, and corresponding counts and percentages for various characterizations within the entity group.

[0177] FIG. **21**A shows an example of a narrative **2102** that can be generated using the conditional outcome framework shown by the upper portions of FIG. **21**A-B as applied to a communication goal statement **2100** of "Analyze the properties by their type" with respect to a data set that includes various properties and associated types for those properties (e.g., single unit homes, duplexes, commercial storefronts, etc.). In this example, the narrative **2102** would be generated after an analysis of the data set arrived at a determination that the outcomes **2104** were true (the size of the group of properties was a "large group" where almost all of the properties in that group shared the same characterization). As can be seen in FIG. **21**A, the narrative text **2102** expresses the following ideas **2106** that are tied to the outcomes **2104**: (1) an identification of the most common type characterization for the properties in the group (single unit homes in this case), (2) the percentage of properties in the group that have this type characterization, and (3) other common type characterizations that exist in the property group.

[0178] FIG. **21**B shows an example of a narrative **2112** that can be generated using the conditional outcome framework shown by the upper portions of FIGS. **21**A-B as applied to the same communication goal statement **2100** shown by FIG. **21**A ("Analyze the properties by their type") but with respect to a different data set that includes various properties and their associated type characterizations. In this example, the narrative **2112** would be generated after an analysis of the data set arrived at a determination that the outcomes **2114** were true (the size of the group of properties was a "decent sized group" where there was a relatively even distribution of properties in that group with respect to their type characterizations). As can be seen in FIG. **21**B, the narrative text **2112** expresses the following ideas **2116** that are tied to the outcomes **2114**: (1) an identification of the common type characterizations for the properties in the group (single family homes, duplex-style homes, and commercial storefronts in this case), (2) the count of properties in the group with each of these common type characterizations, (3) an identification of the uncommon type characterizations for the properties in the group (warehouses and parking lots in this case), and (4) the count of properties in the group with each of these uncommon type characterizations. Thus, FIGS. **21**A and B show yet another example of how the same conditional outcome framework and same communication goal statement can produce dramatically different stories based on the content of the data set under consideration.

Live Story Editing:

[0179] Another innovative feature that may be included in a narrative generation platform is an editing feature whereby a user can use a story outline comprising one or more composed communication goal statements and an ontology to generate a narrative story from source data, where the narrative story can be reviewed and edited in a manner that results in automated adjustments to the narrative generation AI. For example, an author using the system in an editing mode can cause the system to generate a test narrative story from the source data using one or more composed communication goal statements and a related ontology. The author can then review the resulting test narrative story to assess whether the story was rendered correctly and whether any edits should be made. As an example, the author may decide that a different expression for an entity would work better in the story than the expression that was chosen by the system (e.g., the author may decide that a characterization expressed as "slow growth" in the narrative story would be better expressed as "sluggish growth"). The user can directly edit the text of the narrative story using text editing techniques (e.g., selecting and deleting the word "slow" and typing in the word "sluggish" in its place). Upon detecting this edit, the system can automatically update the ontology **320** to modify the subject characterization object **332** by adding "sluggish growth" to the expression(s) **364** for that characterization (and optionally removing the "slow growth" expression).

[0180] To accomplish this, words in the resultant test narrative story can be linked with the objects from ontology **320** that these words express. Further still, sentences and clauses can be associated with the communication goal statements that they serve. In this fashion, direct edits on words, clauses, and sentences by an author on the test narrative story can be traced back to their source

ontological objects and communication goal statements.
[0181] Another example of an innovative editing capability is when an author chooses to re-order the sentences or paragraphs in the test narrative story. Given that sentences and paragraphs in the test narrative story can be traced back to communication goal statements in the story outline, the act of re-ordering sentences and/or paragraphs can cause the system to automatically re-order the communication goal statements in the story outline in accordance with the editing. Thus, consider a story outline that comprises Communication Goal Statement 1 followed by Communication Goal Statement 2 followed by Communication Goal Statement 3 that produces a narrative story comprising Sentence 1 (which is linked to Communication Goal Statement 1), followed by Sentence 2 (which is linked to Communication Goal Statement 2), followed by Sentence 3 (which is linked to Communication Goal Statement 3). If the user decides that the story would read better if Sentence 2 came before Sentence 1, the user can perform this edit in the live story editing mode of the system, and this edit can cause the system to automatically adjust the story outline to comprise Communication Goal Statement 2 followed by Communication Goal Statement 1 followed by Communication Goal Statement 3.
[0182] Similarly, if a user edits the narrative story by deleting a sentence, the system can automatically adjust the story outline by deleting the communication goal statement linked to that sentence.
[0183] Through the automated changes to the ontology **320** and/or story outline, the system can be able to quickly adjust its story generation capabilities to reflect the desires of the author. Thus, during a subsequent execution of the story generation process, the system can use the updated ontology **320** and/or story outline to control the narrative generation process.
[0184] FIGS. **251-273** and their supporting description in Appendix A describe aspects of such editing and other review features that can be included in an example embodiment of a narrative generation platform. Appendix A also describes a number of other aspects that may be included in example embodiments of a narrative generation platform.
[0185] While the invention has been described above in relation to its example embodiments, various modifications may be made thereto that still fall within the invention's scope. Such modifications to the invention will be recognizable upon review of the teachings herein.
TABLE-US-00002 APPENDIX A This appendix describes a user guide for an example embodiment referred to as Quill, and it is organized into the following sections: A1: Introduction A1(i): What is Quill? A1(ii): What is NLG? A1(iii): How to use this Guide A2: Getting Started A2(i): Logging in A2(i)(a): Supported Browsers A2(i)(b): Hosted on-premises A2(ii): General Structure A2(ii)(a): Creating an Organization A2(ii)(b): Creating Users A2(iii): Creating Projects A2(iii)(a): Authoring A2(iii)(b): Data Manager A2(iii)(c): Project Administration A3: Configure a Story from a Blueprint A3(i): Configure a Sales Performance Report A3(i)(a): Headline A3(ii)(b): Overview A3(iii)(c): Drivers A3(iv)(d): Adding Data A3(v)(e): Data Requirements A4: Ontology Management A4(i): Entity Types and Expressions A4(i)(a): Entities Tab A4(i)(b): Creating an Entity Type A4(ii): Relationships A4(ii)(a): Creating a Relationship A4(iii): Characterizations A4(iii)(a): Entity Characterizations A4(iii)(b): Assessment Characterizations A4(iv): Attributes A4(iv)(a): Attribute Values A4(iv)(b): Computed Attributes A5: Configure a Story from Scratch A5(i): The Outline A5(i)(a): Sections A5(i)(a)(1): Renaming a Section A5(i)(a)(2): Deleting a Section A5(i)(a)(3): Moving a Section A5(i)(b): Communication Goals A5(i)(b)(1): Creating a Communication Goal A5(i)(b)(1)(A): Entity Types A5(i)(b)(1)(B): Creating an Entity Type

[0186] The following sections can be read in combination with FIGS. **22-293** for an understanding of how the example embodiment of Appendix A can be used by users.
A1: Introduction
A1(i): What is Quill?
[0187] Quill is an advanced natural language generation (Advanced NLG) platform that transforms structured data into narratives. It is an intelligent system that starts by understanding what the user wants to communicate and then performs the relevant analysis to highlight what is most interesting and important, identifies and accesses the required data necessary to tell the story, and then delivers the analysis in the most intuitive, personalized, easy-to-consume way possible-a narrative.
[0188] Quill is used to automate manual processes related to data analysis and reporting. Its authoring capabilities can be easily integrated into existing platforms, generating narratives to explain insights not obvious in data or visualizations alone.
A1(ii): What is NLG?
[0189] Natural Language Generation (NLG) is a subfield of artificial intelligence (A1) which produces language as output on the basis of data input. Many NLG systems are basic in that they simply translate data into text, with templated approaches that are constrained to communicate one idea per sentence, have limited variability in word choice, and are unable to perform the analytics necessary to identify what is relevant to the individual reader.
[0190] Quill is an Advanced NLG platform that does not start with the data but by the user's intent of what they want to communicate. Unlike templated approaches that simply map language onto data, Quill performs complex assessments to characterize events and identify relationships, understands what information is especially relevant, learns about certain domains and utilizes specific analytics and language patterns accordingly, and generates language with the consideration of appropriate sentence length, structure, and word variability. The result is an intelligent narrative that can be produced at significant scale and customized to an audience of one.
A1(iii): How to Use this Guide
[0191] Getting Started walks through how to log in to Quill and set up Organizations, Users, and Projects. It also provides an overview of the components of Quill.
[0192] Ontology Management is a high-level description of the conceptual elements stories in Quill are based on. This section will help you understand the building blocks of writing a story.

[0193] Configuring a Story from Scratch and Configuring a Story from a Blueprint talk through the steps of configuring a story in Quill. Jump to one of these sections if you want to learn the basics of using Quill.
[0194] Data Management contains the necessary information for setting up data in Quill, discussing the accepted formats and connections.
[0195] Reviewing Your Story discusses the tools available to review, edit, and monitor the stories you configure in Quill.
[0196] Managing Story Versions covers publishing stories and tracking changes made to projects.
[0197] Writing Stories in Production addresses administrative aspects of story generation, including setting up an API endpoint and scheduling story runs.
[0198] Sharing and Reuse goes through how to make components of a particular project available across projects.
[0199] Common Troubleshooting offers simple, easy-to-follow steps for dealing with common questions that arise when working in Quill.
[0200] The Terminology will help you understand the terminology used in this manual and throughout Quill, while the Communication Goal Families describes the available communication goals and how they relate to each other.
[0201] The Miscellaneous section presents an example of a state of Quill functionality.
A2: Getting Started
A2(i): Logging in
A2(i)(a): Supported Browsers
[0202] Quill is a web-based application that supports Firefox, versions 32 ESR and up, and all versions of Chrome. Logging in will depend on whether Narrative Science is hosting the application or Quill has been installed on-premises.
A2(i)(b): Hosted on-premises
[0203] For on-premises installations of Quill, if you are an authenticated user, go to your custom URL to access Quill. You will be taken directly to your project dashboard. If you see an authentication error, contact your site administrator to be set up with access to Quill.
A2(ii): General Structure
[0204] Quill is made up of Organizations and Projects. An Organization is the base level of access in Quill. It includes Administrators and Members and is how Projects are grouped together. Projects are where narratives are built and edited. They exist within Organizations. Users exist at all levels of Quill, at the Site, Organization, and Project levels. Access privileges can be set on a per User basis and apply differently at the Site, Organization, and Project levels. (For more detail, refer to the Permissions Structure section of the Miscellaneous section.)
(ii)(a): Creating an Organization
[0205] Creating an Organization is a Site Administrative privilege. At the time that Quill is installed, whether hosted by Narrative Science or on-premises, a Site Administrator is designated. Only a Site Administrator has the ability to create an Organization (see FIG. **22**).
[0206] Site Administrators can add users, and users can only see the Organizations of which they are members. Site Administrators have access to all Organizations with the View All Dashboards option (see FIG. **23**), but Organization Members do not.
[0207] Members only see the Organizations they have access to in the Organization dropdown and can toggle between them there (see FIG. **24**).
[0208] Site Administrators can use the Organization dropdown to switch between Organizations or from the Organizations page. Each Organization will have a dashboard listing Projects and People.
[0209] FIG. **25** shows where Organization Administrators and Members may create Projects, but only Organization Administrators may create Users. Both Organization Administrators and Members may add Users to Projects and set their permissions. For both Administrators and Members, Quill will show the most recent Organization when first opened.

A2(ii)(b): Creating Users

[0210] Only an Administrator (both Site or Organization) may create a User (see FIG. **26**). Users can be added to Organizations as Administrators or Members (see FIG. **27**).

[0211] Administrative privileges cascade through the structure of Quill. (See Permission Structure in the Miscellaneous section for more information.) That is to say, an Administrator at the Organization level has Administrative privileges at the Project level as well. The Project permissions of Members are set at the Project level.

[0212] At the Project level, a user can be an Administrator, an Editor, or a Reviewer (see FIG. **28**).

[0213] An Administrator on a Project has full access, including all aspects of Authoring, sharing, drafts and publishing, and the ability to delete the Project. An Editor has access to Authoring but cannot share, publish and create a new draft, or delete the Project. A Reviewer only has access to Live Story in Review Mode. A user's access to a Project can be edited on the People tab of the Organization dashboard.

A2(iii): Creating Projects

[0214] Both Administrators and Members can create Projects from the Organization dashboard (see FIG. **29**).

[0215] The creator of a Project is by default an Administrator. When creating a new Project, select from the list of blueprint options whether it will be an Employee History, Empty Project, Municipal Expenses, Network Analysis, or a Sales Performance report (see FIG. **30**).

[0216] This is also where you can access shared components of existing projects which members of an Organization have elected to share for reuse by other Organization members. As shown by FIG. **31**, you can filter them based on what parts of them have been shared: Outline, Ontology, and Data Sources; Outline and Ontology; and Outline. (Refer to the Sharing and Reuse section for additional information.)

[0217] An Empty Project allows the user to configure a Project from the ground up, and a Sales Performance Report provides the framework to configuring a basic version of a sales performance report. A user can be added to a project by clicking the plus symbol within a project (see FIG. **32**) and adding them by user name. To add a user to a Project, the user should be a member of the Organization.

[0218] You can set Project level permissions using the dropdown menu (see FIG. **33**).

[0219] You can edit permissions and remove users here as well (see FIG. **34**).

[0220] Users can also be added to Projects from the People tab of the Organization dashboard (see FIG. **35**).

[0221] Each Project includes Authoring, a Data Manager, and Admin (see FIG. **36**).

[0222] Authoring is where the narrative gets built and refined; the Data Manager is where the data for the story is configured; and Project Administration is where Monitoring, the Change Log, API documentation, Project Settings, and Scheduling are located.

A2(iii)(a): Authoring

[0223] The main view in Authoring is the Outline, as shown by FIG. **37**.

[0224] The Outline is where the narrative is built. Sections can be added to provide structure and organization to the story (see FIG. **38**).

[0225] Communication Goals are then added to a Section (see FIG. **39**).

[0226] Communication Goals are one of the main underpinnings of Quill. They are the primary building blocks a user interacts with to compose a story.

[0227] Authoring is also where Entities are managed (see FIG. **40**).

[0228] An Entity is any primary “object” which has particular Attributes. It can be set to have multiple expressions for language variation within the narrative or have Relationships to other Entities for more complex representations. All of these things comprise an Ontology.

[0229] Data Requirements are how the data that supports a story is mapped to the various story elements.

[0230] Based on the Communication Goals in the Outline, the Data Requirements tab will specify what data points it needs in order to generate a complete story (see FIG. **41**).
[0231] Live Story is a means of reviewing and editing a story generated from the Outline.
[0232] It has two modes, Review mode and Edit mode. Review mode allows the user to see a complete narrative based on specific data parameters (see FIG. **42**). Edit mode allows the user to make changes to the story (see FIG. **43**).
[0233] Drafts and Publishing are Quill's system of managing versions of your story (see FIG. **44**).
[0234] This is how you publish your story configurations and keep a published version as read-only in order to request stories through the API or via the Scheduler. Each Project can only have one draft and one published version at a time.
A2(iii)(b): Data Manager
[0235] The Data Manager is the interface for adding the database connections or uploading the files that drive the story (see FIGS. **45** and **46**).
A2(iii)(c): Project Administration
[0236] The Project Administration features of Quill are Monitoring, the Change Log, API documentation, Project Settings, and Scheduling. They are located in the Admin section of the Project.
[0237] Monitoring allows the user to see the status (success or failure) of generated stories (see FIG. **47**). Stories run through the synchronous API or generated in Live Story will be listed here and can be filtered based on certain criteria (e.g. date, user).
[0238] The Change Log tracks changes made to the project (see FIG. **48**).
[0239] Quill supports on-demand story generation through synchronous API access (see FIG. **49**).
[0240] Project Settings are where you can change the name of the Project and set the project locale (see FIG. **50**). This styles any currencies in your Project to the relevant locale (e.g. Japanese Yen).
[0241] You can set your story to run at regular intervals in Scheduling (see FIG. **51**).
A3: Configure a Story from a Blueprint
[0242] The benefit of configuring a story from a project blueprint is the ability to reuse Sections, Communication Goals, Data Views, and Ontology as a starting point. These blueprints are available in the Create Project screen as discussed in the Getting Started section.
A3(i): Configure a Sales Performance Report
[0243] Select the Performance Project Blueprint and give your project a name. You can always change this later by going to Admin>Project Settings. After the project is created, you'll be taken to Authoring and presented with an Outline that has a "Headline", "Overview", and "Drivers" sections with associated Communication Goals within them (see FIG. **52**).
A3(i)(a): Headline
[0244] To begin, set the Attributes in the Communication Goal in the Headline. Select "the value" (see FIG. **53**) to open a sidebar on the right side of the screen.
[0245] Create an Attribute by entering "sales" and clicking "Create "sales" (see FIG. **54**).
[0246] Then specify "currency" from the list of Attribute types (see FIG. **55**).
[0247] The next step in Attribute creation is to associate the Attribute with an Entity type. Since there are no existing Entity types in this blank Project, you'll have to create one (see FIG. **56**).
[0248] Click "an entity or entity group" to bring out the Entity type creation sidebar (see FIG. **57**).
[0249] Name the Entity type "salesperson" and click to create "salesperson" (see FIG. **58**).
[0250] Set the base Entity type to Person (see FIG. **59**).
[0251] Quill will make a guess at the singular and plural expressions of the Entity type. Make corrections as necessary and click "Okay" (see FIG. **60**).
[0252] There are no designations on the Entity type you created, so click "Okay" to return to the Attribute editing sidebar (see FIG. **61**). A designation modifies the Entity type to specify additional context such as relationships to other Entity types or group analysis.
[0253] Once an Entity type is created, it will be available for selection throughout the project.

Additional Entity expressions can be added in the Entities tab (see Ontology Management).
[0254] Next, you'll specify a Timeframe for the Attribute (see FIG. **62**).
[0255] Click "Timeframe" to create a new Timeframe (see FIG. **63**).
[0256] Choose Month (see FIG. **64**) to complete the creation of the Attribute (see FIG. **65**).
[0257] Click "the other value" to set another Attribute (see FIG. **66**).
[0258] Name it "benchmark" (see FIG. **67**) and set its type to "currency" (see FIG. **68**).
[0259] Associate it with the Entity type "salesperson" and set it to be in the "month" Timeframe (see FIG. **69**).
[0260] Click on the arrow to the left of the Communication Goal in the headline section (see FIG. **70**) to expose the list of related goals.
[0261] The bottom related goal is the Characterization (see FIG. **71**).
[0262] Check the box to opt in to the Characterization (see FIG. **72**).
[0263] Quill has default thresholds to determine the comparative language for each outcome. Entering different values into the boxes (see FIG. **73**), with each value being percentage comparisons calculated against your data view, can change these thresholds (see FIG. **74**). As such, these comparisons are done against numerical Attribute Values. If a value is changed to be less than the upper bound or greater than the lower bound of a different outcome, Quill will adjust the values so that there is no overlap.
A3(ii)(b): Overview
[0264] Configure the first Communication Goal in the Overview section (see FIG. **75**) using the same steps as for the Communication Goal in the Headline section.
[0265] Set the Attribute of the first "Present the value" Communication Goal to be "sales in the month of the salesperson," and the Attribute of the second "Present the value" Communication Goal to be "benchmark in the month of the salesperson" (see FIG. **76**).
[0266] Link the two Present Communication Goals by dragging (using the gripper icon on the right side of the Communication Goal that is revealed when you hover your cursor over the Goal—see FIG. **77**) "Present the benchmark in the month of the salesperson" to overlap "Present the sales in the month of the salesperson" (see FIG. **78**).
A3(iii)(c): Drivers
[0267] Step One: Click "the value" in the first Communication Goal in the Drivers section to set the Attribute. Choose computed value in the Attribute creation sidebar and go into the functions tab in order to select "contribution" (see FIG. **79**).
[0268] Set the Attribute to be "sales" (see FIGS. **80** and **81**).
[0269] Click the first entity and create the new Entity type "sector" of type "Thing" (see FIG. **82**) Add a relationship (see FIG. **83**) and set the related entity as "salesperson" (see FIG. **84**).
[0270] Set the relationship as "managed by" (see FIGS. **85** and **86**).
[0271] Add a group analysis and set the Attribute as "sales" and the Timeframe to "month" (see FIG. **87**).
[0272] Set the second entity to "salesperson" and the timeframe to "month" (see FIG. **88**).
[0273] Step Two: Follow the steps as above to complete the second Communication Goal in the Drivers section but set the position from top to be **2** in the group analysis (see FIGS. **89-90**).
[0274] Step Three: Click into the "Search for a new goal" box and select "Call out the entity" (see FIG. **91**).
[0275] Set the entity to be "highest ranking sector by sales in the month managed by the "salesperson" (see FIG. **92**).
[0276] Then move the goal by grabbing the gripper icon on the right side to the first position in the section (see FIG. **93**).
[0277] Step Four: Create another Call out the entity Communication Goal (see FIG. **94**).
[0278] Create a new Entity type of "customer" and set the base entity type to "thing" (see FIG. **95**).
[0279] Add a group analysis and set the Attribute to "sales" and the Timeframe to "month" (see

FIG. **96**).

[0280] Then add a relationship and set the related entity to be "highest ranking sector by sales in the month managed by the salesperson" and choose the relationship "within" (see FIG. **97**).

[0281] Then move it to the third position in the Drivers section, after the first Present goal (see FIG. **98**).

[0282] Step Five: Create another Call out the entity Communication Goal and set the entity to "second highest ranking sector by sales in the month managed by the salesperson" (see FIG. **99**).

[0283] And move it to the fourth position in the Drivers section, before the second Present goal (see FIG. **100**).

[0284] Step Six: Create another Call out the entity Communication Goal. Create a new entity type of customer following Step Four, but set the related entity to be "second highest ranking sector by sales in the month managed by the salesperson" (see FIG. **101**).

[0285] Step Seven: Finally, create another Call out the entity Goal. Create a new plural Entity type of "regions" and set its type to be "place." Add a group analysis and set the number from top to "3," the Attribute to "sales," and the Timeframe to "month" (see FIG. **102**).

[0286] Then add a relationship, setting the related Entity type as "salesperson" and the relationship as "managed by" (see FIG. **103**).

[0287] The completed outline should match FIGS. **104** and **105**. Quill will update the "Data Requirements" tab with prompts asking for the information necessary to generate the story from that configuration.

A3(iv) (d): Adding Data

[0288] In order to complete the Data Requirements for the story, you add a Data Source to the Project. Go the Data Manager section of the Project to add a Data View (see FIG. **106**).

[0289] Choose to Upload a file and name the Data View (see FIG. **107**). Upload the Sales Performance Data csv file that you were provided.

[0290] Once Quill has saved the Data View to the Project, you will be presented with the first few rows of the data (see FIG. **108**).

A3(v) (e): Data Requirements

[0291] The Data Requirements will guide you through a series of questions to fill out the necessary parameters for Narrative Analytics and Communication Goals (see FIG. **109**). Go to the Data Requirements tab in Authoring.

[0292] See the Data Requirements section of Configure a Story from Scratch for more detail. The completed Data Requirements can appear as shown by FIGS. **110-113**.

[0293] Go to Live Story to see the story (see FIG. **114**).

[0294] Toggles for "salesperson" (see FIG. **115**) and "month" will show you different stories on the performance of an individual Sales Person for a given quarter.

A4: Ontology Management

A4(i): Entity Types and Expressions

[0295] Entity types are how Quill knows what to talk about in a Communication Goal. An Entity type is any primary "object" which has particular Attributes. An example is that a Department (entity type) has Expenses (Attribute)—see FIG. **116**. An Entity is a specific instance of an Entity type, with data-driven values for each Attribute.

[0296] In other words, if you have an Entity type of Department, Quill will express a specific instance of a Department from your data, such as Transportation. Likewise, Expenses will be replaced with the numerical value in your data. Quill also allows you to create Entity and Attribute designations, such as departments managed by the top salesperson or total expenses for the department of transportation (see FIG. **117**).

[0297] When you generate a story with such designations, Quill replaces them with the appropriate calculated values.

A4(i)(a): Entities Tab

[0298] Entity types are managed in the Entities tab (see FIG. **118**).
[0299] Quill defaults to showing all Entity types, but you can filter to only those that are in the story (see FIG. **119**).
[0300] Clicking an Entity type tile allows you to view its details and edit it. Here, you can modify or add Entity expressions (see FIG. **120**), edit or add Entity characterizations (see FIG. **121**), add or edit Attributes associated with the Entity (see FIG. **122**), and add Relationships (see FIG. **123**).
4(i)(b): Creating an Entity Type
[0301] Entity types can be created from the Entities tab (see FIG. **124**) or from the Outline (see FIG. **125**).
[0302] When you create an Entity type, you select its base Entity type from the options of Person, Place, Thing, or Event (see FIG. **126**).
[0303] This gives Quill context for how to treat the Entity. In the case of the Person base Entity type, Quill knows to determine gender and supply an appropriate pronoun.
[0304] Entity types can have multiple expressions. These are managed in the Entities tab of a project (see FIG. **127**).
[0305] They can be added either from the Entities tab (see FIG. **128**) or from Live Story (see FIG. **129**).
[0306] To add expressions, open the details for an Entity type (by clicking on "salesperson," as shown above) and click in the text area next to the plus icon in the sidebar. Type in the expression you want associated with the Entity. You can add expressions for the Specific, Generic Singular, and Generic Plural instances of the Entity by clicking on the arrow dropdown in the sidebar to toggle between the expressions (see FIG. **130**).
[0307] Attributes can be referenced in Specific entity expressions by setting the attribute name off in brackets. For example, if you would like the last name of the salesperson as an expression, set "last name" off in brackets as shown in FIG. **131**.
[0308] You can also opt into and out of particular expressions. If you have multiple expressions associated with the Entity, Quill will alternate between them at random to add Variability to the language, but you can always uncheck the box to turn the expression off (see FIG. **132**) or click on the x icon to remove it completely. You cannot opt out of whichever expression is set as the primary expression, but if you want to make one you've added the primary expression simply click and drag the expression to the top of the list.
A4(ii): Relationships
[0309] Entity types can be tied to each other through Relationships. For example, a City contains Departments, and Departments are within a City (see FIG. **133**). Relationships are defined and created during Entity type creation in Authoring.
[0310] They can also be added to an existing Entity type by editing the Entity type in Authoring. FIG. **134** shows how a relationship can be added from the Entity type tile. FIG. **135** shows setting the related Entity type, and FIG. **136** shows choosing the relationships.
[0311] An Entity type can support multiple relationships. For example, Department has a relationship to City: "within cities"; and a relationship to Line Items: "that recorded line items" (see FIG. **137**).
A4(ii) (a): Creating a Relationship
[0312] If the Relationships already set in Quill do not meet your needs, you can create your own. Type the relationship you want to create in the "search or create" textbox and click "Create new relationship" at the bottom of the sidebar (see FIG. **138**).
[0313] After that, you will be taken through some steps that tell Quill how the new Relationship is expressed. Enter in the present tense and past tense forms of the Relationship, and Quill automatically populates the noun phrase that describes the relationship between the Entities (see FIG. **139**).
[0314] Once you complete the steps for both directions of the relationship (see FIG. **140**), Quill

will apply the relationship to your Entity types and add the relationship to its library. You can use the Relationship again anywhere else in the project.

A4(iii): Characterizations

[0315] Characterizations are editorial judgments based on thresholds that determine the language used when certain conditions are met. Characterizations can be set on Entity types directly or when comparing Attributes on an Entity in a Communication Goal.

A4(iii)(a): Entity Characterizations

[0316] An Entity characterization allows you to associate descriptive language with an Entity type based on the performance of a particular Attribute. For example, you might want to characterize a Sales Person by her total sales (see FIG. **141**).

[0317] Click "+ Characterization" to create a Characterization (see FIG. **142**).

[0318] Once you've named and created the Characterization, you'll have to set the expressions for the Default outcome. Click the grey parts of speech to edit the expression in the sidebar (see FIG. **143**).

[0319] To add an Outcome, click "+Outcome" (see FIG. **144**).

[0320] Change the Outcome label to describe the outcome. For this example, the Outcome label will be "Star" to reflect an exceptional sales performance. Again, edit the expressions by clicking on the grey parts of speech. In order for the outcome to be triggered under specific conditions, you need to add a Qualification (see FIG. **145**).

[0321] Click "+Qualification" to set the value to Sales (see FIG. **146**) and the comparison as "greater than" (see FIG. **147**).

[0322] You have a choice for comparing the value to an Attribute or a static value (see FIG. **148**).

[0323] In this case, choose to keep it a static value and set the value to $10,000 (see FIG. **149**). Follow the same steps to create the lower bound outcome, setting the label as "laggard" and the static value to $1,000 (see FIG. **150**).

[0324] Once you have defined Characterizations on an Entity, you can include them in your story by using the Present the Characterization of the entity Communication Goal (see FIG. **151**).

A4(iii)(b): Assessment Characterizations

[0325] To set the characterizations on a comparative Communication Goal, expand the arrow to the left of the Communication Goal (see FIG. **152**).

[0326] This exposes the list of available subgoals (see section below). At the bottom of this list is a goal to assess the difference between the attributes. Check the box to expose the thresholds applied to the comparison (see FIG. **153**).

[0327] Quill has default thresholds to determine the comparative language for each outcome. These thresholds can be changed by entering different values into the boxes. If a value is changed to be less than the upper bound or greater than the lower bound of a different outcome, Quill will adjust the values so that there is no overlap (see FIG. **154**).

[0328] There is also default language to correspond with each of the possible outcomes. This can also be changed to suit your particular needs and the tone of your story. Click on the green, underlined text to open a sidebar to the right where you can add additional expressions and set which expression you would like to be the primary characterization (see FIG. **155**).

[0329] You can also opt into and out of particular expressions. However, in the example of Appendix A, you cannot opt out of whichever expression is set as the primary characterization. If you have multiple expressions associated with the outcome (see FIG. **156**), Quill will alternate between them at random to add Variability to the language. These additional expressions will be tied to the specific Communication Goal where you added them and will not appear for others. You can also opt into and out of particular expressions, as well as delete them using the x. However, in the example of Appendix A, you cannot opt out of whichever expression is set as the primary expression.

[0330] These expressions can also be edited in Edit mode in Live Story (see FIGS. **157** and **158**).

A4(iv): Attributes

[0331] An Attribute is a data-driven feature on an Entity type. As described above, Quill will express a specified Attribute with the corresponding value in the data based on your Communication Goal. Quill also supports adding modifiers to attributes in order to perform calculations on the raw value in the data.

A4(iv) (a): Attribute Values

[0332] Attribute Values are those values that are taken directly from your data. In other words, no computations are performed on them. An example is the Name of the City. If there is a value in the data for the total expenses of the city, Quill pulls this value directly and performs no computations, unless a data validation rule is applied e.g. "If null, replace with Static Value." which is set in the Data Requirements when mapping the Outline's information needs to your Data View. FIG. **159** shows an attribute creation sidebar. FIG. **160** shows creating an attribute value in the attribute creation sidebar. FIG. **161** shows setting the type of an attribute in the attribute creation sidebar. FIG. **162** shows a completed attribute in a communication goal.

[0333] You also have the option of specifying a Timeframe (see FIGS. **163** and **164**).

[0334] This allows you to restrict the window of analysis to a particular day, month, or year. Create a new Timeframe by selecting one of those three options. Once you've done this, Quill also recognizes the "previous" and "next" instances of that Timeframe (see FIG. **165**). In other words, if you create a day Timeframe, Quill will populate the list of known Timeframes with day, along with previous day and next day.

A4(iv) (b): Computed Attributes

[0335] On the other hand, if the total expenses of the city are calculated by taking the sum of the expenses for each department, Quill allows you to create a Computed Value. Computed Values allow you to compute new values from values in your data and use them for group analysis.

[0336] Computed Values can be aggregations or functions. Aggregations include count, max, mean, median, min, range, total (see FIG. **166**).

[0337] In the example of Appendix A, current functions are limited to contribution, which evaluates how much of an aggregate a component contributed (see FIG. **167**).

[0338] Computed Values can be created from Present or Callout Communication Goals. When you create the attribute you are presenting or using to filter the group of Entities, click into the Computed Value tab to access the list of aggregations and functions.

A5: Configure a Story from Scratch

[0339] Quill allows you to build a story based on an existing blueprint or entirely from the ground up. To build a story specific to your needs, choose to create a Blank Project Blueprint and name it.

A5(i): The Outline

[0340] Once you've created your project, you'll be taken to the Outline (see FIG. **168**).

[0341] The Outline is a collection of building blocks that define an overall Story. This is where you do the work of building your story.

A5(i) (a): Sections

[0342] Create and name Sections to organize your story (see FIG. **169**).

[0343] Once created, a Section can be renamed, deleted, or moved around within the outline.

[0344] Sections are how Communication Goals are grouped together.

A5(i)(a)(1): Renaming a Section

[0345] Click the name of the Section and type in the new name.

A5(i)(a)(2): Deleting a Section Hover your cursor over the Section you want to delete. On the right side, two icons will appear: an ellipses and a gripper icon (see FIG. **170**).

[0346] Click the ellipses to reveal the option to delete the Section (see FIG. **171**).

[0347] If deleted the Section will disappear from the outline along with any Communication Goals it contains.

A5(i)(a)(3): Moving a Section

[0348] As above for deleting a Section, hover your cursor over the Section you want to move. Click and hold the gripper icon (see FIG. **172**) to drag the Section where you want to move it and let go.

A5(i)(b): Communication Goals

[0349] Communication Goals provide a bridge between analysis of data and the production of concepts expressed as text. In other words, they are the means of expressing your data in language.

A5(i)(b)(1): Creating a Communication Goal

[0350] Click the text box where it says to Search for a new goal. Choose the Communication Goal you'd like to use (see FIG. **173**).

A5(i)(b)(1)(A): Entity Types

[0351] Depending on the Communication Goal you choose, you will have to set the Entity type or types it is talking about. An Entity type is any primary "object" which has particular Attributes. An example is that a Department (Entity type) has Expenses (Attribute). An Entity is a specific instance of an Entity type, with data-driven values for each Attribute.

[0352] In the example of the Communication Goal "Call out the entity", the example embodiment for Quill of Appendix A requires that an Entity type be specified. What, in your data, would you like to call out? Click "the entity" in the Communication Goal to open a sidebar to the right (see FIG. **174**).

[0353] Here you can select among Entity types that already exist or create a new one. Available entities include entities created from the outline or the entities tab (including any characterizations).

A5(i)(b)(1)(B): Creating an Entity Type

[0354] Click "new" in the Entity sidebar (see FIG. **175**). Then choose from existing Entity types or create a new one. Set whether the Entity type is singular or plural (see FIG. **176**). Once you have created the Entity type, you will be asked to set its base Entity type: Event, Person, Place, or Thing (see FIG. **177**). Next, set the plural and singular expressions of the Entity type (see FIG. **178**). Quill takes an educated guess at this, but you have the opportunity to make changes. Next you will designate any relationships, group analysis, or qualification pertaining to the Entity type (see FIG. **179**).

[0355] Quill lets you know the state of an Entity type, whether it is unset, in progress, or valid based on the appearance of the Entity type in the Communication Goal. The Entity type appears grey when unset (see FIG. **180**), blue when being worked on (see FIG. **181**), and green when valid (see FIG. **182**).

[0356] Adding a relationship allows you to tell Quill that an Entity is related to another Entity. To do so, choose to Add Relationship as you create your Entity type. Then set or create the Entity type that this Entity has a relationship to (see FIG. **183**). Quill suggests a number of relationships from which you can choose, including "lives in", "managed by", "within", and more. FIG. **184** shows a list of available relationships between two entities (department and city). FIG. **185** shows an entity with a designated relationship. You can also create Relationships that will be added to the library.

[0357] When creating an Entity type of the base type event (see FIG. **186**), Quill will prompt you to set a timeframe for it to associate the event with (see FIG. **187**).

A5(i)(b)(1)(C): Creating a Relationship

[0358] If the Relationships already set in Quill do not meet your needs, you can create your own. Type the relationship you want to create in the "search or create" textbox and click "Create new relationship" at the bottom of the sidebar (see FIG. **188**).

[0359] After that, you will be taken through some steps that tell Quill how the new Relationship is expressed. Enter in the present tense and past tense forms of the Relationship, and Quill automatically populates the noun phrase that describes the relationship between the Entities (see FIG. **189**).

[0360] Once you complete the steps for both directions of the relationship (see FIG. **190**), Quill will apply the relationship to your Entity types and add the relationship to its library (see FIG. **191**). You can use the Relationship again anywhere else in the project.

[0361] You can also apply Group Analysis to an Entity type (see FIG. **192**).
[0362] In the example of Appendix A, rank is supported. This allows you to specify which Entity in a list of Entities to use in a Communication Goal. Select whether you are asking for the position from the top or the position from the bottom and the ranking of the Entity you want (see FIG. **193**). FIG. **194** shows setting the attribute to perform the group analysis by. FIG. **195** shows an Entity type with group analysis applied.
[0363] You also have the option of specifying a Timeframe (see FIG. **196**).
[0364] This allows you to restrict the window of analysis to a particular day, month, or year (see FIG. **197**).
[0365] Create a new Timeframe by selecting one of those three options. Once you've done this, Quill also recognizes the "previous" and "next" instances of that Timeframe (see FIG. **198**). In other words, if you create a day Timeframe, Quill will populate the list of known Timeframes with day, along with previous day and next day.
[0366] Once you have completed the steps to create an Entity type, Quill adds it to the list of Entity types available for use throughout the story. In other words, you can use it again in other parts of the Outline.
A5(i)(b)(1)(D): Characterizations
[0367] Characterizations are editorial judgments based on thresholds that determine the language used when certain conditions are met. Characterizations can be set on Entity types directly or when comparing Attributes on an Entity in a Communication Goal.
[0368] Refer to Characterizations in Ontology Management for more information on Entity Characterizations.
[0369] To set the characterizations on a comparative Communication Goal, expand the arrow to the left of the Communication Goal (see FIG. **199**).
[0370] This exposes the list of available subgoals (see section below). At the bottom of this list is a goal to characterize the difference between the attributes. Check the box to expose the thresholds applied to the comparison (see FIG. **200**).
[0371] Quill has default thresholds to determine the comparative language for each outcome. These thresholds can be changed by entering different values into the boxes. If a value is changed to be less than the upper bound or greater than the lower bound of a different outcome, Quill will adjust the values so that there is no overlap (see FIGS. **201** and **202**).
[0372] There is also default language to correspond with each of the possible outcomes. This can also be changed to suit your particular needs and the tone of your story. Click on the green, underlined text to open a sidebar to the right where you can add additional expressions and set which expression you would like to be the primary expression (see FIG. **203**).
[0373] If you have multiple expressions associated with the outcome (see FIG. **204**), Quill will alternate between them at random to add Variability to the language. These additional expressions will be tied to the specific Communication Goal where you added them and will not appear for others. You can also opt into and out of particular expressions, as well as delete them using the x. However, you cannot opt out of whichever expression is set as the primary expression.
A5(i)(b)(2): Deleting a Communication Goal
[0374] To delete a Communication Goal, hover your cursor over it to reveal a trash can icon (see FIG. **205**). Click it to delete the Communication Goal.
A5(i)(b)(3): Moving a Communication Goal
[0375] Moving a Communication Goal is done the same way as moving a Section. Hover your cursor over the Communication Goal to reveal the gripper icon (see FIG. **206**).
[0376] Click and move the Communication Goal within the Section or to another section (see FIG. **207**). Be careful when you move Communication Goals to make sure there is space between them.
[0377] Communication Goals without space between them are Linked Goals, described below.
A5(i)(b)(4): Linked Goals

[0378] Quill supports linking Communication Goals. This allows the user to express ideas together. For example, you may wish to talk about the number of departments in a city along with the total budget for the city. Hover your cursor over the Communication Goal to reveal the gripper icon, click and drag it above the goal you wish to link (see FIG. **208**). They will always be unlinked by revealing the gripper icon again by hovering, and moving the Communication Goal into an empty space on the Outline.

[0379] When you link the Communication Goal that expresses the number of departments and the Communication Goal that expresses the total budget for the city (see FIG. **209**), Quill will attempt to express them together with smoother language such as combining them into one sentence with a conjunction.

A5(i)(b)(5): Related Goals (Subgoals)

[0380] Some goals support related goals, or subgoals. This allows you to include supporting language without having to create separate Communication Goals for each related idea. For example, if you have a Communication Goal comparing attributes on an entity—in this case, the budget and expenses of the highest ranking department by expenses within the city—you may also wish to present the values of those attributes. Expand the Communication Goal to expose those related goals and opt into them as you like (see FIG. **210**).

A5(i)(b)(6): Styling Communication Goals

[0381] Quill allows for styling Communication Goals for better presentation in a story. Hover your cursor over a Communication Goal to reveal the "Txt" dropdown on the right side (see FIG. **211**).

[0382] Here, you can choose whether the language expressed is styled as a headline (see FIG. **212**), normal text (see FIG. **213**), or bullets (see FIG. **214**).

A5(i)(b)(7): Charts

[0383] Charts are supported for two Communication Goals: Present the [attribute] of [a group] and Present the [attribute] of a [group of events]. For either of these goals, to get a chart, go to the Txt dropdown and select Chart (see FIG. **215**).

[0384] This will render the Communication Goal as a chart.

[0385] Present the [attribute] of [a group] (see FIG. **216**) will result in a bar chart (see FIG. **217**).

[0386] Present the [attribute] of [a group of events] (see FIG. **218**) will result in a line chart (see FIG. **219**).

A5(i) (c): Data Requirements

[0387] Once you have configured your story, Quill will ask where it can find the data to support the Entity types and Attributes you have specified in the Communication Goals. Go to the Data Requirements tab in Authoring to provide this information (see FIG. **220**).

[0388] The Data Requirements will guide you through a series of questions to fill out the necessary parameters for Narrative Analytics and Communication Goals. For each question, select the data view where that data can be found and the appropriate column in the table.

A5(i) (c) (1): Tabular Data

[0389] FIG. **221** shows an example where the data is tabular data.

A5(i) (c) (2): Document-Based Data

[0390] FIG. **222** shows an example where the data is document-based data.

[0391] Where the value supplied is numerical, Quill will provide analytic options for cases where there are multiple values (see FIG. **223**). "Sum" sums values in a column like a Pivot Table in a spreadsheet. "Constant" is if the value does not change for a particular entity. For example, the quarter may always be Q4 in the data.

[0392] For each Entity type, Quill will ask for an identifier (see FIG. **224**).

[0393] This is what Quill uses to join data views. An identifier has no validation options as it doesn't actually appear in the story. (Data Validation is discussed below.)

[0394] The final question in Data Requirements will be to identify the main Entity the story is about (see FIG. **225**).

[0395] In the city budget example, Quill needs to know what city the story will be about. This can be set as a static value (e.g. Chicago) or as a Story Variable (see FIG. **226**).
[0396] A Story Variable allows you to use a set of values to trigger stories. In other words, if your data contains city budget information for multiple cities, setting the city the story is about as a Story Variable will allow you to run multiple stories against the same dataset. The location of the value for the Story Variable is defined earlier in Data Requirements where Quill asks where to find the city.
[0397] If there is a Timeframe in the Headline of the story, Quill will need you to identify this in Data Requirements as well.
[0398] As with the entity, this can be a static value or a Story Variable. It can also be set as the run date (see FIG. **227**), which will tell Quill to populate the value dynamically at the time the story is run. (See the Scheduling section for more information.)
[0399] A5(i) (d): Data Formatting
[0400] Quill allows you to set the format for certain data points to have in your data source so it can be mapped to your Outline. These formats are set based on the ontology (Entities, Attributes, etc.) being used in your Communication goals, with default styling applied to values. See the Miscellaneous section for specific styling information. As you configure the appropriate data formats present in your data view, validation rules can be applied if the types do not match for a particular story run. For example, if Quill is expecting the expenses of a city to be a currency and receives a string, the user is provided with various options of actions to take. These are specified in the Data Validation section below. To select the format of any date fields you may have, go to the Data Requirements tab in Authoring and click the checkbox icon next to a date (see FIG. **228**) to pull out the sidebar (see FIG. **229**).
[0401] Click on the date value to open a list of date format options and make your selection (see FIG. **230**).
A5(i) (e): Data Validation
[0402] Quill supports basic data validation. This functionality can be accessed in Data Requirements. Once you specify the location of the information in the data, a checkbox appears next to it. Click this to open the data validation sidebar (see FIG. **231**).
[0403] You will be presented with a number of options in a dropdown menu for what to do in the case of a null value (see FIG. **232**).
[0404] You can tell Quill to fail the story, drop the row with the null value, replace the null value with a value you provide in the text box below, or ignore the null value.
A6: Data Management
[0405] Quill allows for self-service data management. It provides everything you need to upload files and connect to databases and API endpoints.
A6(i): Getting Data Into Quill
[0406] Quill supports data in tabular or document-based formats. Tabular data can be provided to Quill as CSV files or through table selections made against SQL connections (PostgreSQL, Mysql, and Microsoft SQL Server are supported). Document-based data can be provided by uploading a JSON file, creating cypher queries against Neo4j databases, a MongoDB connection, or through an HTTP API connection (which you can also set to elect to return a CSV).
A6(i) (a): Uploading a File
[0407] You can upload a CSV or JSON file directly to Quill in the Data Manager. In the Views tab, choose to Upload a file from the Add a Data View tile (see FIG. **233**).
[0408] Provide the name of the view and upload the file. The amount of time it will take to upload a file depends on the size of the file for a maximum file size of 50 MB, and operating against a data base connection is recommended. This automatically populates the Source Name. FIG. **234** shows an example where a CSV file is uploaded. FIG. **237** shows an example where a JSON file is uploaded. You can edit the Source Name, which is helpful when file names are difficult to parse

and for readability when selecting the file from the Live Story dropdown when previewing your story. Quill automatically detects whether the data is in tabular or document form and samples a view of the first few rows or lines of data. FIG. **235** shows an example of uploaded tabular data, and FIG. **236** shows a sample view of tabular data. FIG. **238** shows an example of uploaded document-based data, and FIG. **239** shows a sample view of document-based data.

[0409] Quill also supports uploading multiple data sources into one Data View. This functionality can be accessed in the Data View by clicking the three dots icon (see FIG. **240**).

[0410] Here, you can upload additional files or add additional connections (see FIG. **241**). If you have multiple data sources in a Data View, you can set a source as primary, edit, or delete it. New data files or tables can be added to an existing data view, but only tabular sources can be added to tabular views and document-based sources to document-based views. To make the newly uploaded source your primary dataset, click on the three dots icon and select it as primary. This makes it the file used during runtime story generation requests or Live Story previews.

A6(i)(b): Adding a Connection

[0411] You can also provide data to Quill by connecting to a SQL database, a cypher query against a Neo4j database, a MongoDB database, or an HTTP API endpoint. You can add a connection from the Data View tab by choosing Start from Connection from the Add a Data View tile (see FIGS. **242** and **243**) or by choosing to Add a Connection from the Connections tab (see FIG. **244**).

[0412] Quill will ask for the appropriate information to set up each type of connection. FIG. **245** shows an example of credentials for a SQL database connection. FIG. **246** shows an example of credentials for a Neo4j database connection. FIG. **247** shows an example of credentials for a MongoDB database connection. FIG. **248** shows an example of credentials for an HTTP API connection.

[0413] The connection will be made, subject to network latency and the availability of the data source. Data Views from connections are made from the Views tab. Choose Start from a Connection and select the connection you created (see FIG. **249**).

[0414] Quill will prompt you to specify the table to add the data source. For neo4j connections, you will have to put in a cypher query to transform the data into tabular form (see FIG. **250**). From there, Data Requirements can be satisfied using the same experience as tabular and document-based views allowing for type validation rules to be set as needed.

A7: Reviewing Your Story

[0415] Once you have configured your story with Sections and Communication Goals, and satisfied the Data Requirements against a data source, you can review or edit its contents, understand the logic Quill used to arrive at the story, and monitor the status of stories you run.

A7(i): Live Story

[0416] Live Story is where you can see the narrative expression of the story you configured in the Outline (see FIG. **251**).

[0417] If you have set up your story to be based on Story Variables (as opposed to a static value), you can toggle between them (see FIG. **252**) and see how the narrative changes.

[0418] You can also switch between data sources (see FIG. **253**).

[0419] Click the “rewrite” button to generate a new narrative to see how any additional expressions you have added affect the Variability of the story (see FIG. **254**).

[0420] Live Story has two modes: Edit and Review.

A7(i) (a): Edit Mode

[0421] Edit mode allows you to make changes to the language in your story (see FIG. **255**).

A7(i)(a)(1): Entity Expressions

[0422] You can add Entity expressions from Live Story (in addition to the Entities tab). If you click on any Entity (highlighted in blue under the cursor) (see FIG. **256**), a sidebar will open on the right side (see FIG. **257**).

[0423] You can add Entity expressions by typing in the area next to the plus sign. You can also opt into and out of particular expressions. If you have multiple expressions associated with the Entity, Quill will alternate between them at random to add Variability to the language. Click the rewrite button to see how your story changes. As described in the Ontology Management section, you can also click, hold, and drag an expression to the top of the list and opt out of the additional expressions to set it as primary.

A7(i)(a)(2): Characterization Expressions

[0424] You can edit the expressions in any Characterizations you have set on Compare Communication Goals from Edit mode in Live Story. As with Entity expressions, Characterization expressions will be highlighted in blue when you move the cursor over them (see FIG. **258**).

[0425] Click on the expression to open a sidebar to the right where you can add additional expressions and set which expression you would like to be the primary expression (see FIG. **259**).

[0426] Quill will alternate between them at random to add Variability to the language. These additional expressions will be tied to the specific Communication Goal where you added them and will not appear for others. You can also opt into and out of particular expressions, as well as delete them using the x. However, you cannot opt out of whichever expression is set as the primary expression. See Assessment Characterizations in Ontology Management for more detail.

A7(i)(a)(3): Language Guidance

[0427] You can add set Language Preferences, such as word order choice, to your story in the Edit mode of Live Story using Language Guidance. Hover over a section (sections correspond to Sections in the Outline) of the story to reveal a Quill icon on the right side (see FIG. **260**).

[0428] Click it to isolate the section from the rest of the story (see FIG. **261**).

[0429] Click on a sentence to expose any additional expressions you can opt into (see FIG. **262**).

[0430] Quill generates expressions using language patterns appropriate to the Communication Goal, so the number of additional expressions will vary and not all sentences will have additional expressions. Quill will alternate between them at random to give your story more language variation.

A7(i)(b): Review Mode

[0431] Project Reviewers have access to this aspect of Authoring. In review mode (see FIG. **263**), you can read stories and switch datasets to see how they affect the story. You can also see if there are any errors in the story with Quill's logic trace (discussed below).

A7(ii): Logic Trace

[0432] Quill allows you to see the steps it takes to express Communication Goals as a story. If you click on any sentence in the story in Live Story in Review mode, Quill will show the underlying Communication Goal or Goals (see FIG. **264**).

[0433] Expand the arrow on the left of the Goal to see the steps Quill took to retrieve data based on the Communication Goal and Data Requirements (see FIG. **265**).

[0434] In this case, it created a Timeframe and an Entity Type. Then it “shows its work” of pulling the Attribute Value of “sales” constrained by the Timeframe of “month” and associated with the Entity Type “Salesperson 1.”

[0435] The Logic Trace can also be downloaded as a JSON file from the Monitoring tab in Admin (see FIG. **266**).

A7(iii): Monitoring

[0436] You can monitor the status of any stories you run, whether they were written in Live Story or generated through API requests in the Monitoring tab in Admin. Here, you can see whether stories succeeded or failed, and filter for specific stories using the available filters below (see FIG. **267**).

[0437] Use the Newer and Older buttons to scroll through the stories (see FIG. **268**), and use the arrows on the column headers to set search criteria. You can filter by story status (see FIG. **269**), when the story completed writing (see FIG. **270**), the user who requested the story (see FIG. **271**), a run type for the story (see FIG. **272**), and a version for the story (see FIG. **273**).

A8: Managing Story Versions

[0438] Quill supports creating and keeping track of changes to and versions of the stories you configure.

A8(i): Drafts and Publishing

[0439] Once you have configured your story and are satisfied with its expression in Live Story, you can Publish the draft of your story (see FIG. **274**).

[0440] Once Published, your story will go live and that version will be the one that Quill uses when stories are requested through an API connection. After a draft has been Published, any changes you wish to make to the Project should be made after creating a new draft (see FIG. **275**).

[0441] Once a new draft has been created, it can be deleted. You can also switch to the Published version if you want to abandon the changes you have made in the new draft. The drafts and publishing dropdown is also where you can save the Project as a blueprint to share with others in the Organization (see FIG. **276**). This is discussed in Sharing.

[0442] Project Administrators are the only ones with draft creation and publishing privileges. While Editors may make changes to active drafts, they cannot publish them or create new ones. Reviewers only have access to review mode in Live Story and cannot create, make changes to, or publish drafts.

A8(ii): Change Log

[0443] Quill tracks configuration changes made within a Project. Anytime a user makes a change or adds a new element to a Project, it's noted in the Change Log. The Change Log can be accessed in the Admin section of Quill (see FIG. **277**).

[0444] Here, you can see a list of all changes in the Project, the users that made the changes, the date and time the changes were made, and the version of the project the changes were made to. As with Monitoring, you can page through the list of changes by clicking on the Newer and Older buttons (see FIG. **278**).

[0445] The Time, User, and Version information can be used to filter the list by using the drop-downs next to the column headers. FIG. **279** shows an example dropdown to filter by time.

[0446] FIG. **280** shows an example dropdown to filter by user. FIG. **281** shows an example dropdown to filter by version.

[0447] You can also download the changes made as a CSV (see FIG. **282**) in order to plot the Project activity or aggregate it for purposes of visualization or archiving.

A9: Writing Stories in Production

A9(i): API

[0448] Quill supports on-demand story generation by connecting to an API. The documentation can be accessed from Admin.

[0449] API request samples are available in the API Documentation tab of the Admin section of Authoring (see FIG. **283**). These samples are based on the project Outline configuration and available data source connections. Parameters and output formatting can be set here so that stories can be requested to meet specific data requirements from an outside application.

[0450] The Request Builder allows the user to select the dataset, set the format (Plain Text, HTML, JSON, or Word) of the output, and choose the syntax of the request sample (see FIG. **284**).

[0451] An external application can use the sample to post requests to the API to generate stories from Quill once the text in red has been replaced with its specific variables (see FIG. **285**). Each Quill user will be able to request a certificate and key from their system administrator.

A9(ii): Scheduling

[0452] Stories can also be run on a schedule (see FIG. **286**).

[0453] Once Scheduling is enabled (see FIG. **287**), stories can be run at scheduled intervals (see FIG. **288**) beginning at a specific date and time. The run can be ended at a specific time or continue indefinitely. Additionally, you can set the format of the story to Plain Text, HTML, or JSON (see FIG. **289**), which can then be retrieved for viewing from the Monitoring page. Published Project schedules are un-editable at this time. To edit the schedule, create a new draft and update as needed.

A10: Sharing and Reuse

[0454] Projects can be shared with other users. The Draft dropdown menu includes an option to Save as Blueprint (see FIG. **290**).

[0455] Here, you can give the shared version of the Project a name and description (see FIG. **291**).

[0456] You can also specify how much of the Project you make available for sharing. You can include the Outline, Ontology (Entities), and Data Sources, the Outline and Ontology, or just the Outline (see FIG. **292**).

[0457] Projects that have been saved as blueprints can be accessed when choosing a blueprint. Quill defaults to including all shared projects, but you can filter blueprints based on what elements they include (Outline, Ontology, Data Sources) (see FIG. **293**).

A11: Terminology

[0458] The following provides a glossary for various terms used in connection with describing the example embodiment of Appendix A.

[0459] An Organization is a collection of Projects managed by an Administrator. Members of an Organization have access to those Projects within it that they have permissions for. Outlines are collections of building blocks that define an overall Story.

[0460] Communication Goals provide a bridge between analysis of data and the production of concepts expressed as text.

[0461] Narrative Analytics generate the information needed by Communication Goals to generate stories.

[0462] Projects are where stories are configured. A Project includes Authoring, the Data Manager, and Admin.

[0463] Project Blueprints are templates comprised of an Outline, specific story sections, and collections of Communication Goals.

[0464] An Ontology is a collection of Entity Types and Attributes, along with their expressions, that powers how Quill expresses your story.

[0465] An Entity Type is any primary “object” which has particular Attributes. An example is that a Sales Person (entity) has Sales (attribute). Relationships provide context for entities within a story.

[0466] Every Entity Type has a Base Entity Type that identifies to Quill whether it is a Person, Place, Thing, or Event.

[0467] Computed Values are a way of reducing a list of values into a representative value. The currently available aggregations are count, maximum, mean, median, minimum, and total, and the currently available function is contribution.

[0468] Characterizations are editorial judgments based on thresholds that determine the language used in communication goals when certain conditions are met.

[0469] Expressions are the various words Quill uses to express a particular concept generated by the combination of executing Narrative Analytics and Story Elements.

[0470] A Timeframe is a unit of time used as a parameter to constrain the values included in the expression of a Communication Goal or story.

[0471] Variability is variation in the language of a story. Variability is provided through having multiple Entity and Characterization expressions as well as option into additional sentence expressions through Language Guidance.

[0472] Authoring includes the Outline, Data Requirements, and Live Story. This is where you configure Communication Goals, map Entity Types and Attributes to values in the data, and review generated stories.

[0473] Data Requirements are how a user tells Quill the method by which we will satisfy a Communication Goal's data requirements. These are what a Narrative Analytic and Communication Goal need to be able to express a concept. These are satisfied either directly by configuration of the data requirements or through the execution of Narrative Analytics.

[0474] A Story Variable is the focus of a story supplied at runtime as a value from a data source (as opposed to a static value).

[0475] A Draft is an editable version of the story in a Project. Project Administrators and Editors have the ability to make changes to Drafts. Project Administrators can publish Drafts and create new ones.

[0476] The Data Manager is the part of the Project where Data Views and Data Sources backing the story are managed. This is where files are uploaded and database connections are added.

[0477] A Data View is a used by Quill to map the Outline's information needs against Data Sources. A Project can be backed by multiple Data Views that are mapped using Identifiers in the schemas.

[0478] A Data Source is a file or table in a database used to support the Narrative Analytics and generation of a story.

[0479] Admin allows you to manage all aspects of story generation other than language and data. This is where Monitoring, the Change Log, API Documentation, Project Settings, and Scheduling are located.

A12: Communication Goal Families

[0480] The example embodiment of Appendix A supports three communication goal families: Present, Callout, and Compare.

Present

[0481] The Present goal family is used to express an attribute of a particular entity or group of entities.

[0482] Most Present goal statements have the form “Present the attribute (or computed value) of the specified entity/group.” For example: [0483] Present the price of the car. [0484] Present the price of the highest ranked by reviews item. [0485] Present the average value of the deals made by the salesperson.

[0486] The two exceptions to this form are when the Count or Contribution computed values are used, in which case the statements look like this: [0487] Present the count of the group. [0488] E.g. Present the count of the franchises in the region. [0489] Present the attribute contribution of the entity to the parent entity. [0490] E.g. Present the point contribution of the player to the team.

Callout

[0491] The Callout goal family is used to identify the entity or group of entities that has some editorially-interesting position, role, or characteristics. E.g. the highest ranked salesperson, franchises with more than $1k in daily sales, players on the winning team, etc. Every Callout goal statement has the same structure: “Callout the specified entity/group.” For example: [0492] Callout the highest ranked by sales salesperson. [0493] Callout the franchises with more than 1,000 in daily sales. [0494] Callout the players on the winning team.

Compare

[0495] The Compare goal is used to compare the values of two attributes on the same entity. Every Compare goal has the same structure: Compare the first attribute of the specified entity to the second attribute. For example: [0496] Compare the sales of the salesperson to the benchmark. [0497] Compare the final value of the deal to the expected value. [0498] Compare the revenue of the business to the expenses.

A13: Miscellaneous

A13(i): Charts

[0499] Quill is able to express certain configured goals as Charts, such as Bar and Line. These have default styling and colors and are guided by the Communication Goal's Narrative Analytics. Charts are supported in each available output format.

A13(ii): Supported Document Structures

[0500] Generally, Quill supports documents that are homogenous (uniformly structured) with stable keys. Example permutations of supported structures are described below.

A13(ii) (a): Single Document

[0501] In this example, as long as all documents contain the same keys (in this case, “a”, “b”, and “c”) Quill can use this data structure.

TABLE-US-00003 { “a”: 1, “b”: 2, “c”: 3 }

A13(ii) (b): Nested Documents

[0502] Documents with other documents nested within them are supported, though the nested documents must be homogenous with stable keys across documents.

[0503] A first example is:

TABLE-US-00004 { “a”: { “aa”: 1, “ab”: 2 }, “b”: { “ba”: 3, “bb”: 4 } }

[0504] A second example is:

TABLE-US-00005 [{ “a”: 1, “b”: [{ “ba”: 11, “bb”: 12 }, “ba”: 20, “bb”: 44 }] }]

A13(ii) (c): Unsupported Structures

[0505] The example embodiment of Appendix A does not support heterogeneous documents (non-uniform) or documents where values are used as keys.

TABLE-US-00006 { “1/1/1900”: “45”, “1/2/1900”: “99”, “1/3/1900”: “300” }

A13(iii): Styling Rules

Oxford Commas

[0506] Quill does not use Oxford commas. So it writes like “Mary spoke with Tom, Dick and Harry” and not like “Mary spoke with Tom, Dick, and Harry.”

Spaces Between Sentences

[0507] Quill puts one space between sentences.

Dates

[0508] Year: Datetimes that are just years are expressed numerically. [0509] 2016->“2016” [0510] 1900->“1900”

[0511] Month and Year: Datetimes that are just months and years have written out months and numeric years. [0512] 2016-03->“March 2016” [0513] 2015-**11**->“November 2015”

[0514] Day, Month, and Year: Datetimes that are full dates are written out months with numeric days and years. [0515] 2016-03-25->“Mar. 25, 2016” [0516] 2015-11-05->“Nov. 5, 2015”

Percents

[0517] Percents are rounded to two places, trailing zeros are removed, and a “%” is appended. [0518] 53.2593->“53.26%” [0519] 53.003->“53%”

Ordinals

[0520] Ordinals are written with numerical contractions. [0521] 1->“1st” [0522] 2->“2nd” [0523] 3->“3rd” [0524] 556->“556th”

Decimals

[0525] Decimals are written out with decimal parts and commas inserted. [0526] 1.1->“1.1” [0527] 1.9->“1.9” [0528] 123456789->“123,456,789”

Currencies

[0529] Currencies are currently assumed to be USD. In the future, they can be locale-specific (e.g. Euros). They're styled differently based on how big they are.

Less than One Thousand

[0530] Rounds to two decimal places. There are always two decimal places. [0531] 3->“$3.00” [0532] 399.9999->“$400.00”

Less than Ten Thousand [0533] Rounds to an integer. [0534] 5000.123->“$5,000” [0535] 4171->“$4,171”

Less than One Million [0536] Rounds to thousands with zero decimal places, appends a “K” [0537] 500,000->“500K” [0538] 123,456.789->“123K”

Less than One Billion

[0539] Rounds to millions with one decimal place if necessary, appends an “M” [0540]

500,000,000->“500M” [0541] 500,100,000.12->“500.1M”
Less than One Trillion
[0542] Rounds to billions with two decimal places if necessary, appends an “M” [0543] 500,000,000,000->“500B” [0544] 500,100,000,000.12->“500.1B” [0545] 500,130,000,000.12->“500.13B”
Supported Datetime Formats
[0546] The following datetime formats are supported in Quill. [0547] 01/31/15 [0548] 01/31/2015 [0549] 31-Jan-2015 [0550] Jan. 31, 2015 [0551] Tuesday, Jan. 31, 2015 [0552] Tuesday, Jan. 31, 2015, 01:30 AM [0553] 2015-01-31T01: 30:00-0600 [0554] 20150131 [0555] 2015-01-31 13:30:00 [0556] 1/31/2015 01:30:45 [0557] 31-01-2015 01:30:45 [0558] 1/31/2015 1:30:45 [0559] 1/31/2015 01:30:45 AM [0560] 31/01/2015 01:30:45 [0561] 2015/01/31 01:30:45
A13(iv): Using Multiple Data Views
[0562] Users can satisfy their outline's data requirements using multiple data views. While it may often be more straightforward to create a de-normalized view in the source database, the following use cases are supported. These apply to both tabular and document-based data sources.
Single Entity Type, Attribute Lookup by Entity ID
[0563] Quill can return the Gender from Data View 2 associated with the Sales Person's ID in Data View 1 using the Sales Person ID.
TABLE-US-00007 Data View 1 Sales Person ID Sales Person Name 123 Aaron Young 456 Daisy Bailey
TABLE-US-00008 Data View 2 Sales Person ID Gender 123 Male 456 Female
Two Entity Types
[0564] Quill can match the Transactions in Data View 2 to the Sales People in Data View 1 by Sales Person ID.
TABLE-US-00009 Data View 1 Sales Person ID Sales Person Name 123 Aaron Young 456 Daisy Bailey
TABLE-US-00010 Data View 2 Transaction ID Amount Sales Person ID 777 $100.00 123 888 $70.00 456 999 $20.00 123
A13(v): Permission Structure
TABLE-US-00011 Quill Access Create Create API Create Role Organizations Users Token Projects Site X X X X Administrator Organization X X X Administrator Organization X X Member
TABLE-US-00012 Project Access Live Create Live Story: and Story: Add Edit Edit Publish Review Role Users Story Mode Drafts Mode Administrator X X X X X Editor X X X Reviewer X

Claims

1. A method for generating a narrative from structured data according to a narrative generation process, the structured data comprising a plurality of data values associated with a plurality of data attributes, the method comprising: receiving user input via a communication interface, the user input identifying a communication goal for analyzing one or more data attributes for a specified group calculation for a specified entity group via a model parameterized with respect to the structured data; receiving output of the model, wherein the output includes (1) a value determined for the specified group calculation based on the structured data and (2) a summary of a subset of the structured data selected automatically based on the value to determine information to include in the narrative; and displaying a natural language narrative about the structured data to a client machine via a graphical user interface, the natural language narrative being generated based on the output, wherein the natural language narrative expresses the summary in natural language and satisfies the communication goal.
2. The method recited in claim 1, the method further comprising: transmitting an instruction to determine a parameterized model based on the received user input.

3. The method recited in claim 1, wherein the graphical user interface also includes a data chart generated based on the structured data, the data chart supporting the communication goal.
4. The method recited in claim 1, wherein the user input is received via the graphical user interface, the communication goal being one a plurality of communication goals that may be specified via the graphical user interface.
5. The method recited in claim 1, wherein the communication goal for analyzing the specified group calculation with respect to the structured data is identified based on data comprising a communication goal statement.
6. The method recited in claim 1, the method further comprising: displaying a graphical input interface for allowing a user to specify the communication goal, the communication goal being one of a plurality of communication goals that may be specified via the graphical input interface, the user input being provided via the graphical input interface.
7. The method recited in claim 6, the method further comprising: select the model to be parameterized from among a plurality of models based on the communication goal for analyzing the specified entity group with respect to a specified attribute.
8. The method recited in claim 6, wherein the communication goal is identified based on data comprising a communication goal statement.
9. The method recited in claim 1, the method further comprising selecting a plurality of idea data structures within the model based on the specified group calculation, wherein each selected idea data structure corresponds to a different idea and has an associated condition; and evaluating conditions of the selected idea data structures based on the structured data to determine which of the selected idea data structures is applicable to the structured data and thus serves as a basis for the summary.
10. The method recited in claim 1, wherein the model comprises a plurality of conditional outcome data structures corresponding to different characterizations for a distribution of members of the specified entity group with respect to values of the one or more data attributes.
11. A system configured to perform a method for generating a narrative from structured data according to a narrative generation process, the structured data comprising a plurality of data values associated with a plurality of data attributes, the system comprising: a communication interface configured to receive a user input identifying a communication goal for analyzing one or more data attributes for a specified group calculation for a specified entity group via a model parameterized with respect to the structured data; and a processor configured to receive output of the model, wherein the output includes (1) a value determined for the specified group calculation based on the structured data and (2) a summary of a subset of the structured data selected automatically based on the value to determine information to include in the narrative, the processor configured to provide a natural language narrative about the structured data to a client machine via a graphical user interface, the natural language narrative being generated based on the output, wherein the natural language narrative expresses the summary in natural language and satisfies the communication goal.
12. The system recited in claim 11, wherein the processor is further configured to transmit an instruction to determine a parameterized model based on the received user input.
13. The system recited in claim 11, wherein the graphical user interface also includes a data chart generated based on the structured data, the data chart supporting the communication goal.
14. The system recited in claim 11, wherein the user input is received via the graphical user interface, the communication goal being one a plurality of communication goals that may be specified via the graphical user interface.
15. The system recited in claim 11, wherein the communication goal for analyzing the specified group calculation with respect to the structured data is identified based on data comprising a communication goal statement.
16. One or more non-transitory computer readable media having instructions stored thereon for

performing a method for generating a narrative from structured data according to a narrative generation process, the structured data comprising a plurality of data values associated with a plurality of data attributes, the method comprising: receiving user input via a communication interface, the user input identifying a communication goal for analyzing one or more data attributes for a specified group calculation for a specified entity group via a model parameterized with respect to the structured data; receiving output of the model, wherein the output includes (1) a value determined for the specified group calculation based on the structured data and (2) a summary of a subset of the structured data selected automatically based on the value to determine information to include in the narrative; and displaying a natural language narrative about the structured data to a client machine via a graphical user interface, the natural language narrative being generated based on the output, wherein the natural language narrative expresses the summary in natural language and satisfies the communication goal.

17. The one or more non-transitory computer readable media recited in claim 16, the method further comprising: transmitting an instruction to determine a parameterized model based on the received user input.

18. The one or more non-transitory computer readable media recited in claim 16, wherein the graphical user interface also includes a data chart generated based on the structured data, the data chart supporting the communication goal.

19. The one or more non-transitory computer readable media recited in claim 16, wherein the user input is received via the graphical user interface, the communication goal being one a plurality of communication goals that may be specified via the graphical user interface.

20. The one or more non-transitory computer readable media recited in claim 16, wherein the communication goal for analyzing the specified group calculation with respect to the structured data is identified based on data comprising a communication goal statement.