DOWNLOAD PDF SCENARIO VISUALIZATION

Chapter 1 : WOA1 - Scenario modeling and visualization - Google Patents

The MIT Press is a leading publisher of books and journals at the intersection of science, technology, and the arts. MIT Press books and journals are known for their intellectual daring, scholarly standards, and distinctive design.

These scenarios are modeled by aggregating events into activities that are then aggregated, themselves, into scenarios. A scenarios analyzer accesses data logs to extract and analyze data for the modeled scenario. The analyzed data is visualized as a histogram with roll up and drill down functionality. Some computer systems are relatively large and have a variety of different types of data collected for them, so that a user or administrator or other person can monitor the information being processed by, or performance of, the computer system. Business systems can include, for instance, customer relations management CRM systems, enterprise resource planning ERP systems, line-of-business LOB systems, among others. Such business systems perform workflows and processes, and generate user interface displays that allow users to interact with the business systems. The users can do this in order to perform activities or tasks, to carry out their business. Telemetry and analytics are a part of many data driven business and engineering processes in various kinds of software and services. This data can be aggregated to identify scenario indicator metrics, such as key performance indicators or KPIs. The indicator metrics are then used to compare scenario usage, performance, or reliability across different versions and demographics. Summarization and aggregation techniques are used to generate the indicator metrics, and various pivots and filters are enabled so that a user can drill down to view more detailed data, when the metrics indicate that a problem may exist with a given scenario. These types of aggregations assume a parametric distribution of the data. However, the telemetry data may be from varied segments of the population, or be influenced by other variables, and this can contribute to the data being multi-modal or non-parametric. Thus, when the data is compared across populations, it can generate false positive or negative KPI indications, which add noise to the telemetry and analytics systems, and can render the entire report non-actionable. However, these efforts have proven very expensive in terms of computing overhead and labor. However, scenario usage often changes over time due to the nature of the software business. Thus, even if the aggregations are tuned to the specific usage, pre-aggregation tuning needs to be done separately for each pivot value. Most of the pivot values e. Therefore, it can be very expensive both in terms of computation and query resources to support these indicator metrics across a range of pivots, in such a way that they can be actionable indicator metrics. A scenario analyzer accesses data logs to extract and analyze data for the modeled scenario. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter. The claimed subject matter is not limited to implementations that solve any or all disadvantages noted in the background. Architecture shows a business system , from which telemetry data collection system collects telemetry data and stores it in telemetry data store Architecture also illustratively includes data analysis and visualization system System obtains data from telemetry data store and generates various visualizations of that data for various scenarios that can be modeled by a user. Before describing the overall operation of architecture in more detail, a number of the items shown in architecture will be described. It could be an engineering system or another computer system for which telemetry data is to be collected and analyzed. Data store , itself, illustratively includes entities , workflows , processes and it can include other business data records or other data as well. For instance, a vendor entity describes and defines a vendor. A customer entity describes and defines a customer. A business opportunity entity describes and defines a business opportunity. These are only a small number of the various entities that can be defined within business system Business processing component illustratively accesses workflows and processes to run applications on the various entities in order to perform the business operations for the business that is deploying business system In doing so, user interface component illustratively generates user interface displays that can have user input mechanisms for interaction by user The user illustratively interacts with user input mechanisms in order to interact with, and manipulate, business system A simplified example of a scenario is to load a given form, such as a customer form. User may routinely need to view various customer forms and enter or review data on those forms. Thus, one common scenario for user may be to load the customer form. The scenario also has an end point at which the form is loaded and rendered to the user. In between the start and end points, various other events and activities can be performed by system For instance, system can access data store to obtain the form. It can then access the data store again to load data into the form. Each of these events or activities may, themselves, have a start and end point and a duration. Thus, the "load customer form" scenario may be defined by a plurality of different events, activities which can be a sequence of events , and they can generate a variety of different types of data, such as the start time, the end time, the duration, the frequency with which the event, activity or scenario was performed, etc. Telemetry agent can be a distributed agent service, or it can be deployed in another way. It illustratively collects telemetry data from business system The telemetry data can be a wide variety of different types of data, such as events and the information that defines the events such as start time, count, duration, end time, etc. Uploader component can be embodied as a scalable uploading service, or it can be embodied in other ways. It uploads telemetry data to telemetry data store and stores it, in one embodiment, as event logs It can store it in other ways as well. Scrubber component accesses event logs and scrubs the data such as by removing noisy data, placing the data in a given, expected format, etc. Scenario modeling system generates a modeling user interface display with modeling user input mechanisms for interaction by user In one embodiment, user can be the same as user , and this is indicated by arrow In another embodiment, however, the two user are different. Scenario modeling is described in greater detail below with respect to Figures F. Scheduler component schedules analysis runs for the various scenarios that are modeled by scenario models Analyzer component obtains data from processed events logs in data store and runs the analysis for the various scenario models In one embodiment, analyzer component generates one or more histograms for each of the key performance indicator metrics corresponding to a given scenario, and the histograms can be used for comparing indicator metrics. This makes the computation map-reducible and reporting additive in nature. The analyzed data, for the various modeled scenarios, is then stored in scenario analysis data store as scenario data Performing the analysis to generate scenario data is described in greater detail below with respect to Figure 3. Visualization systems can be a wide variety of different types of clients, such as spread sheet clients, business intelligence clients, database management services, etc. Scenario visualization illustratively includes data and user input mechanisms A user which can be the same as, or different from, users and can interact with user input mechanisms to perform explorations on the data presented by scenario visualization The user can actuate these input mechanisms to perform drill down and roll up functionality to see more or less detailed information on the given visualization. In addition, user input mechanisms can be a wide variety of other user input mechanisms as well, such as pivot functions, filters, etc. Generating the visualization is described in greater detail below with respect to Figures B. A scenario is a set of events and activities with which make up a logical or business scenario of interest. Events and activity instances for a scenario are tied together by a parent scenario identifier, which can be a correlating identifier, a timestamp, or a machine name, among other things. Each scenario can include temporal activities and sub-activities as well as spatial data which provides contextual details for different instances of a given scenario. Each activity can have a duration metric and other associated metrics, such as counts, types, etc. Activities and spatial data are defined based on events which are raw instrumentation events. The parent scenario identifier is used to tie events belonging to each instance of a given scenario. A child scenario identifier is used to track related and child scenarios that are spawned from a parent scenario. This is indicated by block in Figure 2. This can take a wide variety of different forms. For instance, the user can provide authentication information such as a username and password , or it can be done in other ways Receiving the set of identified events and data points is indicated by block in Figure 2. The particular events and data points that can be used by scenario modeling system to model a scenario will vary based upon the particular system where the scenarios are being modeled. System can obtain these events and data points in a wide variety of different ways as well. For instance, in one embodiment, scenario modeling system automatically obtains the events and data points from business system This is indicated by block In another embodiment, system can receive user inputs defining the various events and data points, from a user, an administrator, etc. System can receive the set of events and data points that can be used to model scenarios in other ways as well, and this is indicated by block System then receives user inputs selecting events to define activities within the scenario. System then receives user inputs selecting fields e. The metadata entities that are defined in order to configure a scenario include events which are raw events from the monitored system e. Data points include spatial and temporal metric points in the event. Activities are combinations of one or more events that determine a functional or idle state of the system, a single end event, along with a time indicator that indicates a time in a state, or begin and end events. Metrics can include key performance indicators KPIs that are calculated based on data points or durations. The metrics can also include transformations which can be expressed over data points to aggregate at a per scenario instance granularity, or a different granularity. The metrics can include quantization methods for generating histograms and the metrics can be reported as histogram dimensions with a single frequency count as the measure. The user then provides visualization inputs defining the visualization for this scenario, as indicated by block A visualization includes a configuration for exploring and reporting the various metrics that are defined for the scenario. Figure 2A shows one example of a user interface display that lists a set of events that can be used to construct a scenario. In the embodiment shown in Figure 2A, a list of events are provided with an event name in column and a creation date in column The events can be listed in other ways as well. These events can be selected and ordered to model a scenario. Display includes an event name for a given event indicated generally by number

Chapter 2 : Role-based Scenario Visualization | AllChange

These views allow you to visualize a scenario while the scenario is being edited, while the scenario executes, and after the scenario is executed. Several settings allow you to control the content and appearance of the views.

Code map Visualize the organization and relationships in existing code. To identify classes, their relationships, and their methods, create a code map that shows those elements. Artifacts can be many things, such as namespaces, projects, classes, methods, and so on. Layers represent and describe the roles or tasks that the artifacts perform in the system. You can also include layer validation in your build and check-in operations to make sure that the code stays consistent with its design. To keep the code consistent with the design, Dinner Now and Lucerne use the following dependency diagram to validate their code as it evolves: Dependency diagram for Dinner Now integrated with Lucerne The layers on this diagram link to the corresponding Dinner Now and Lucerne solution artifacts. For example, the Business layer links to the DinnerNow. Business namespace and its members, which now include the PaymentApprover class. The Resource Access layer links to the DinnerNow. The arrows, or dependencies, specify that only the Business layer can use the functionality in the Resource Access layer. As the teams update their code, layer validation is performed regularly to catch conflicts as they occur and to help the teams resolve them promptly. The teams work together to incrementally integrate and test the two systems. They first make sure that PaymentApprover and the rest of Dinner Now work with one another successfully before they deal with PaymentProcessing. The following code map shows the new calls between the Dinner Now and PaymentApprover: Code map with updated method calls After they confirm that the system works as expected, Dinner Now comments out the PaymentProcessing code. The layer validation reports are clean, and the resulting code map shows that no more PaymentProcessing dependencies exist: Code map without PaymentProcessing A dependency diagram has the following major features: Layers describe logical groups of artifacts. A link is an association between a layer and an artifact. To draw new layers and then link them to artifacts, use the toolbox or right-click the diagram surface to create the layers, and then drag items to those layers. The number on a layer shows the number of artifacts that are linked to the layer. These artifacts can be namespaces, projects, classes, methods, and so on. When you interpret the number of artifacts on a layer, remember the following: If a layer links to an artifact that contains other artifacts, but the layer does not link directly to the other artifacts, then the number includes only the linked artifact. However, the other artifacts are included for analysis during layer validation. For example, if a layer is linked to a single namespace, then the number of linked artifacts is 1, even if the namespace contains classes. If the layer also has links to each class in the namespace, then the number will include the linked classes. If a layer contains other layers that are linked to artifacts, then the container layer is also linked to those artifacts, even though the number on the container layer does not include those artifacts. To see the artifacts that are linked to a layer, right-click the dependency, and then click View Links to open Layer Explorer. A dependency indicates that one layer can use the functionality in another layer, but not vice versa. A bidirectional dependency indicates that one layer can use the functionality in another layer, and vice versa. To display existing dependencies on the dependency diagram, right-click the diagram surface, and then click Generate Dependencies. To describe intended dependencies, draw new ones.

Chapter 3 : Scenario Visualization FLAMES Modeling & Simulation software by Ternion

In order to solve problems, humans are able to synthesize apparently unrelated concepts, take advantage of serendipitous opportunities, hypothesize, invent, and engage in other similarly abstract and creative activities, primarily through the use of their visual systems.

The Evolution of the Visual System and Scenario Visualization 29 5 Scenario Visualization, Creative Problem Solving, and Evolutionary Psychology References Index Acknowledgments Throughout the construction of this book, I learned firsthand that philosophy and the academic world, in general, are social enterprises. I could not have completed this book without so many helpful and rigorous people. George Terzis, the director of my doctoral dissertation and philosophical mentor at Saint Louis University in St. Louis, Missouri, was most helpful and most rigorous. The amount of time, energy, care, wisdom, and concern he gave to me will never be forgotten. He is a true philosopher, a scholar, and a gentleman. I spent as much time discussing this project with Brian Cameron. Terzis, I owe the majority of my philosophical formation to Brian Cameron, who, besides having one of the keenest analytical minds, probably was the most critical of my project every step of the way. I thank him for these criticisms and have tried responding to many of them in this book. Another friend of knowing is Kevin Decker, who got me to thinking about issues in naturalism and evolution in the first place. What wisdom in me that was not gained from Dr. Terzis and Brian, I received from Kevin. Given the interdisciplinary nature of my project, I have tried to make sure that all of the information in this book is accurate. Other top-notch philosophers, neuroscientists, biologists, and psychologists at Saint Louis University, Washington University in St. Louis, University of Missouriâ€"St. Louis, and Southwest Minnesota State University took the time out of their busy schedules to look over earlier versions of this book and give me comments. In particular, I have learned a lot from Dr. I hand drew all of the illustrations in this book and, with the kind help of a computer graphic designer in St. Louis named Chuck Hart, was able to convert them into electronic format. With his calming demeanor, Tom Stone, Philosophy Acquisitions Editor at MIT Press, was a relaxing influence for me during the book-writing process, and I thank him for believing in the project in the first place. I also want to acknowledge the dedication of referees, copy editors, proofreaders, copy setters, and other folks in the publication process who work for MIT. I would like to thank Kim Sterelny, Anthony Freeman, Vincent Colapietro, Raymond Russ, and other editors of the following journals for the permissions to utilize sections of my previously published work in this book: Finally, my wife, Susan Marie Arp, has looked over parts of my work and has given me comments to think about throughout this project including my very first substantial criticism. I have the most profound respect for her, she continues to be my anchor, and I dedicate all of my projects to her. Routine Problem Solving versus Nonroutine Creative Problem Solving Imagine being a dentist in the early part of the nineteenth century. Now, imagine going to the dentist to have a tooth pulled in the early part of the nineteenth century. In those days, pulling teeth was a painful experience for the patient, as there were no known anesthetics in use at the time. Such were the methods that Dr. Horace Wells likely used to solve the problem of pain associated with tooth extractions while working as a dentist in Hartford, Connecticut, circa These methods probably were nothing new, and we can imagine that dentists had been using these remedies for some time so as to alleviate or prevent the pain associated with tooth extraction. One evening in , Dr. Wells attended an amusing public demonstration of the effects of inhaling a gas called nitrous oxide with his friend, Samuel Cooley. Cooley volunteered to go up on stage to inhale the gas, and he proceeded to do things like sing, laugh, and fight with other volunteers who had inhaled the gas. As a result of the high jinks, Cooley received a deep cut in his leg before coming back to his seat next to Dr. Upon witnessing this event, a light went on in Dr. In fact, over the next several years Dr. Wells proceeded to use nitrous oxide and was successful at painlessly extracting teeth from his patients, and the seeds of modern anesthesia were sown Roberts, We can imagine Dr. In such a case, we would have an instance of what Mayer has called routine problem solving, whereby a person recognizes many possible solutions to a problem given that the problem was solved through one of those solutions in the past. However, humans also can engage in activities that are more abstract and creative, such as invent tools based upon mental blueprints, synthesize concepts that, at first glance, seemed wholly disparate or unrelated, and devise novel solutions to problems. Wells decided to use nitrous oxide with his patients, he pursued a wholly new way to solve the problem of pain. This was an instance of what Mayer has called nonroutine creative problem solving, which involves finding a solution to a problem that has not been solved previously. The introduction of nitrous oxide in order to extract teeth painlessly would be an example of nonroutine creative problem solving because Dr. Wells did not possess a way to solve the problem already, and he had not pursued such a route in the past. Not only do people make insightful connections like that of Dr. Wells but they take advantage of serendipitous opportunities, invent products, manufacture space shuttles, successfully negotiate environments, hypothesize, thrive, flourish, and dominate the planet by coming up with wholly novel solutions to problemsâ€”primarily through the use of their visual systems. How is this possible? In this book, I give an evolutionary account of the human ability to solve nonroutine, vision-related problems creatively in their environments. I argue that, by the introduction of the Upper Paleolithic toolmaking industry near the close of the Pleistocene epoch, our hominin species evolved a conscious creative problem solving capacity I call scenario visualization that enabled individuals to fashion the tools and other products necessary to outlive other hominin species and populate the planet. Scenario visualization is a conscious activity whereby visual images are selected, integrated, and then transformed and projected into visual scenarios for the purposes of solving problems in the environments one inhabits. The evidence for scenario visualization is found in the kinds of complex tools our hominin ancestors invented in order to survive in the ever-changing environments of the Pleistocene world. In this book, Routine Problem Solving versus Nonroutine Creative Problem Solving 3 I also argue that this conscious capacity shares an analogous affinity with neurobiological processes of selectivity and integration in the visual system, namely, processes that enable animals to select relevant information from environmental stimuli and to organize this information in a coherent way for the animal. Further, I show that similar processes of selectivity and integration can be found in the activities of organisms in general. Because the brain is an evolved organ, a complete explanation of these processes and capacities must appeal to general biological and evolutionary principles. The evolution of these processes in our hominin past, I argue, helps account for the modern-day conscious ability of humans to utilize visual information so as to solve vision-related, nonroutine problems creatively in the environments they inhabit. Principally, I am a philosopher of mind and biology, and, insofar as this is the case, I am concerned with two basic questions concerning human nature, namely, What are humans, in essence, that distinguishes them from the rest of reality? The hypothesis of scenario visualizationâ€”as one form of conscious activityâ€”and its emergence in an evolutionary history are my small attempts to answer these fundamentally philosophical questions. Of course, I will not answer these questions completely. I concur with Pinker , p. The ultimate goal of my project is to explain how humans evolved a specific kind of conscious, vision-related, creative problem solving ability I call scenario visualization also see Arp, a, b, a, a, c. I do this in order to offer a philosophy of biology that is comprehensive enough to account for the levels of biological phenomena that are relevant to my project, and the upshot is to lay the groundwork for showing that there is an analogous continuity of operation in the biological world, ranging from the activities of organelles in a cell to the complex workings of neural networks in a brain from which conscious abilities emerge also Arp, b. Besides being organized in such a way as to produce homeostasis in the organism, the processes in which the components of the organism are engaged possess certain properties. These properties include abilities to exchange data internally, selectively convert data to information, integrate that information, and process information from environments also Arp, a. Having established these properties in the first chapter, in the second chapter I put forward what I call the homeostatic organization view HOV of organisms, whereby the components of organisms are organized to function so as to maintain the homeostasis of the organism at the various levels in the hierarchy Arp, a. Because of HOV, starting with the organelles that make up a cell and continuing up the hierarchy of systems and processes in an organism, we can maintain that there are clear instances of emergent biological phenomena. Using HOV, I endorse a form of what is known as nomological emergence in the metaphysical realm. Since the endorsement of a set of entities in the metaphysical realm requires an adequate description of those entities, I argue that it may be useful for a researcher to think like an as-if realist when describing the traits and processes of organisms also see Arp, c, d. Whereas I use HOV to give credence to a version of nomological emergence in the metaphysical Routine Problem Solving versus Nonroutine Creative Problem Solving 5 realm, I use as-if realism to give credence to a corresponding form of representational emergence in the epistemological realm. The end result is a better understanding of the epistemological views that underpin my metaphysical views in philosophy of science and philosophy of biology. In fact, it is essential to my project that I explain and defend a description of functions because my hypothesis concerning scenario visualization depends upon certain functional mechanisms of the mind having evolved to solve specific problems encountered in various Pleistocene environments also Arp, b. I show how these accounts can complement and be made compatible with one another. Building upon the work of the first two chapters, in the third chapter I show how the subsystems and processes associated with vision in mammals comprise a hierarchically organized system exhibiting similar, analogous kinds of properties of information exchange, selectivity, and integration found in all organisms also Arp, b, a. My analysis of the brain is restricted to the primary processes and mechanisms associated with the mammalian visual system for three reasons. Second, the visual system is present in many kinds of vertebrate species thought to be homologous to human beings. And third, the visual system plays a central role in the evolutionary account I give of the progression from noncognitive visual processing to conscious cognitive visual processing in terms of scenario visualization. As I go on to demonstrate, visual processing is an important factorâ€”if not the most important factorâ€”in the evolution of conscious creative problem solving capacities in humans. In the third chapter, I also distinguish four levels of visual processing in animals. The first is a noncognitive visual processing that takes place at 6 Introduction the lowest level of the visual processing hierarchy associated with the eye and its neural projections to the lateral geniculate nucleus and primary visual cortex. The second is a cognitive or psychological visual processing that occurs at a higher level in the visual hierarchy associated with the what and where unimodal areas of the brain. The third is a cognitive visual processing that occurs at an even higher level in the visual hierarchy whereby visual unimodal areas are integrated in the visual unimodal association area of the brain. The fourth is a conscious cognitive visual processing that occurs at the highest level of the visual hierarchy whereby the visual association areas are integrated with other sensory modalities, the limbic areas, and frontal areas of the brain also Arp, a, b. By the end of the third chapter, I show that the visual systems of mammals, in general, function so as to produce visual cognition. Special attention is paid to visual modularity and visual integration. Visual modularity refers to the fact that the visual system is made up of distinctly functioning and interacting modules or areas having evolved to respond to certain features of an object in typical environments. I do this utilizing the anatomical evidence from fossils and living species thought to be homologous to ancient species. I also use archeological evidence from ancient toolmaking techniques, since I believe that the evolution of tool-types parallels the evolution from noncognitive visual processing, through cognitive visual processing, to conscious cognitive visual processing. The variety and complexity of tools discovered and dated by archeologists offer us compelling evidence that the brain and visual system have evolved with the passage of time also Arp, a, c. Routine Problem Solving versus Nonroutine Creative Problem Solving 7 I suggest that advanced forms of toolmaking require scenario visualization, a conscious activity whereby visual images are selected, integrated, and then transformed and projected into visual scenarios for the purposes of solving problems in the environments one inhabits also see Arp, a, b, a, a, c. As a conscious process, scenario visualization is distinct from the cognitive processes of simply forming a visual image or recalling a visual image from memory; these activities can be performed by nonhuman primates, mammals, and certain other animals. Scenario visualization requires a mind that is more active in the utilization of visual images through the processes of selectivity, integration, and projection into future scenarios. It is not the having of visual images that is important; it is what the mind does in terms of actively selecting and integrating visual information for the purposes of solving some problem relative to some environment that really matters. In this project, I am concerned mostly with the progression from cognitive visual processing to conscious cognitive visual processing, the relationship of these processes to one another, and, ultimately, how conscious cognitive visual processingâ€”in terms of scenario visualizationâ€”evolved from cognitive visual processing. There is a huge amount of literature devoted to

DOWNLOAD PDF SCENARIO VISUALIZATION

questions about the existence of psychological phenomena and whether psychological phenomena supervene upon or emerge from neurobiological phenomena for starters, see Chalmers, ; Heil, a, b; Arp, b, d. Working out the problems associated with these issues constitutes solving several so-called mindâ€”body problems. Now, no one has been able to give a satisfactory account of how it is that psychological statesâ€”particularly conscious psychological statesâ€—arise from, as well as interact with, the gray matter of the brain. Although I will not be able to completely solve the mindâ€”body problem of how it is that conscious experience can emerge from and interact with the gray matter of the brain, my hypothesis concerning scenario visualization is an attempt to explain one aspect of our consciousness and the reason for its emergence in our species. Given that modern humans evolved from early hominins, I further fortify the emergence of scenario visualization by presenting the psychological evidence that this kind of activity occurs in our species when trying to solve certain problems, as well as by presenting the 8 Introduction neurobiological evidence showing that our brains are wired so that this kind of psychological activity can occur in the first place also Arp, a, c. We are the only kind of species that can scenario visualize, and what I suggest by the end of the fourth chapter is threefold. First, modern-day humans have the unique ability to actively select and integrate visual images from mental modules so as to transform and project those images in visual scenarios for the purposes of negotiating environmentsâ€”this is scenario visualization.

Chapter 4 : Scenario: Change your design using visualization and modeling - Visual Studio | Microsoft Doc

SMS - Scenario Visualization Return.

Chapter 5 : Scenario Visualization: An Evolutionary Account of Creative Problem Solving - PDF Free Dowr

Chapter 6 : SMS - Scenario Visualization

Scenario Visualization has 5 ratings and 0 reviews. An account of how humans evolved a conscious, vision-related ability unique to their species in order.

Chapter 7 : Scenario visualization : an evolutionary account of creative

Scenario: Change your design using visualization and modeling. 11/04/; 20 minutes to read Contributors. all; In this article. Make sure that your software system meets users' needs by using the visualization and modeling tools in Visual Studio.

Chapter 8 : Visualization of country scenarios - DDPP

Scenario visualization is a conscious activity whereby visual images are selected, integrated, and then transformed and projected into visual scenarios for the purposes of solving problems in the environments one inhabits.

Chapter 9 : Scenario Visualization: An Evolutionary Account of Creative Problem Solving by Robert Arp

If a picture is worth a thousand words, a data visualization is worth at least a million. As inspiration for your own work with data, check out these 15 data visualizations that will wow you. Taken together, this roundup is an at-a-glance representation of the range of uses data analysis has, from.