**Mellanox Call Center**

**Data Analyzer**

Anan Fakheraldin 302629969, Benjamin-Alex Filin 310819230

Supervisor: Dr. Malka Grossman

**Abstract*.*** *In our project Mellanox Data Analyzer we employ methods of machine learning to build a workflow to troubleshoot a problem with a network of Mellanox Technology's customer, based on previous similar errors in company database and what had been done to solve them, analyze the errors on time and configurational scale to advice on threshold adaptation to prevent future network issues and false alarms and providing more efficient day-to-day and long term work environment.*

**Keywords***: machine learning, Natural Language Processing, question answering machine, knowledge based artificial intelligence*

1. INTRODUCTION

1.1. Mellanox overview:

Mellanox Technologies is a leading supplier of end-to-end InfiniBand and Ethernet interconnect solutions and services for servers and storage.

Mellanox offers a choice of fast interconnect products: adapters, switches, software and silicon that accelerate application runtime and maximize business results for a wide range of markets including high performance computing, enterprise data centers, Web 2.0, cloud, storage and financial services.

1.2. Mellanox Call Center:

Mellanox Call Center provides permanent manned support services for Mellanox customers throughout the year (7/24/365) based on the appropriate support contract.

The Call Center is the customer’s point of contact (mostly for emergency cases) for initiating and advancing technical related and support administrative issues.

Call Center activities are designed to accomplish the following targets:

* Provide systematic and efficient call handling services to Mellanox customers.
* Maintain information flow according to developing situations.
* React to cases according to predefined procedures.

1.3. Mellanox Care – Proactive Care Platform:

24/7 fabric management services provided by Mellanox networking experts Mellanox Care services use a combination of advanced monitoring software with a 24/7 human expert service of Mellanox personnel.

Mellanox Care will identify – Keep alive Events that will alert and address hardware failures, non-optimal configuration, service degradation issues and more. Above and beyond keeping the fabric healthy, and addresses the more complex performance issues and bottleneck scenarios, which are impacting application performance.

The current status is as follows:

* All logs are examined manually by call center
* All work flows and troubleshooting are based on representative’s experience
* Existing, reoccurring and resolved errors are verified manually

Mellanox Data analyzer objectives to day to day work are as follows:

* To analyze the Mellanox care Daily report - provide more work efficient environment.
* To timeline and analyze all previous site cases to identify reoccurring issue on port \ switch per customer.
* To synchronize and monitor any frequent alarms\ triggers to locate reoccurring issues with Mellanox products on all Mellanox care Sites
* To provide a day \ week statistics - threshold adaptation advise for each site.

The aim of this project is upon receiving a network error from a customer to analyze and suggest a work flow to troubleshoot a certain network error using a machine learning algorithms (knowledge based AI).

1.4 Machine learning algorithm implementation:

Finding all correspondences with this error, retrieve comments in the correspondences between support engineers and customers from the company's database.

Identify helpful and unhelpful engineer’s advices from the input by the customer, using Natural Language Processing (NLP) algorithms, and “score” the engineer’s replies.

Suggest what should be the next response by the engineer based on high scoring answers and prior actions leading to fault isolation.

2. Background and Related Work:

2.1 Mellanox resources:

2.1.1 Customer Resource Management system:

Mellanox's online Customer Resource Management (CRM) system – SalesForce, provides a comprehensive online tool to manage all of the customer's support issues in one place.

1. Complete case management including reporting support issues and tracking their progress
2. A searchable knowledge database to find solutions, best practices and worthy information.

3. Access to documentation and drivers/firmware/software downloads

4. Built-in RMA (Return Merchandise Authorization) request and tracking system.

2.1.2. Case management and workflow:

Mellanox support engineers handle each customer request according to suitable procedure designed according to the following parameters:

Case type:

* Support cases are handled according to case priority.
* RMA requests are handled by first line support accompanied by the relevant departments.

Case priority:

* Priority 1- Procedure for urgent\ fatal issues set to establish communication and urgency for customer’s issue. Meant to ensure that proper attention of Mellanox functions, Service-level agreement is met and to raise the awareness of related account teams and other professional groups (OPS, PM, AE, OEM account team, etc.) when dealing with a fatal network or production down situation (showstopper).
* Priority 2- Procedure for medium severity issues.
* Priority 3- Procedure for low severity issues.

Priority 2 and Priority 3 cases are handled by Pre-defined layers support system, the first line support start handling this type of cases and make the first contact with the customer and according to urgency and the complexity of the issue the case can be escalated to higher level.

Customer support coverage:

According to Mellanox policy each customer should get the support he needed therefor five types of coverage were defined:

* No support contract - Cases from account with no support contract should be handled by the system administrator only.
* Bronze support coverage - The Bronze support program is Mellanox's basic level support package, tailored for system administrators that are self-sufficient in supporting their Mellanox infrastructure. Support engineers provide the customer with troubleshooting steps for hardware components to insure that customer’s hardware is working properly with the latest available software.
* Silver support coverage – The Silver support program is Mellanox's most popular support package and it provides complete end-to-end support for Mellanox solutions. First line support handles this cases according to priority 2 and priority 3 procedures.
* Gold customer - The Gold support program is the Mellanox premium service program for mission critical deployments where a small percentage of down time could result in a significant loss to businesses. Cases from gold customers escalated automatically to second support tier, in order to insure proper attention of Mellanox functions.

2.1.3. Unified Fabric Manager (UFM®) Software for Data Center Management:

Mellanox’s Unified Fabric Manager (UFM®) is a powerful platform for managing scale-out computing environments. UFM eliminates the complexity of fabric management, provides deep visibility into traffic and optimizes fabric performance.

UFM covers many areas such as fabric monitoring, provisioning, device management, Subnet Management and more.

2.1.4. Mellanox Care - Daily Report:

Mellanox Daily Report is a snapshot of the environment it is running on that collects all the relevant information on Mellanox products installed.

It contains the following reports:

* UFM Health Report
* Fabric Health report

The content of the reports is regarded as trade secret to intellectual property of Mellanox Technologies.

2.1.5. Mellanox Care Policy Booklet:

Mellanox Care Booklet list all possible events captured by Mellanox Care application

For each event you will find the following attributes:

* Description – Unique problem description that distinguish each error.
* Code – Internal parameter that is related to the case description.
* Category (Device management, Fabric management, etc.) – This indicate which layer is effected by this error.
* Related items/Potential root cause – Potential hardware/software failure indicator.
* Description of error – Detailed problem that can be provided to the customer.
* First Line Support corrective actions – Troubleshooting steps to solve the issue.
* Support type/process (Standard/Fatal).

2.1.6. Mellanox Resource limitations:

Data from SalesForce cannot be stored locally as this information is classified, thus we cannot build our own Database and neither could we save a rating for each solution we find. For every query the learning algorithm must be activated again.

Loading and analyzing all case comments might take a long time – the purpose is to be as efficient as possible

2.2 Introduction to Machine learning:

2.2.1 Case-Based Reasoning:

The ability to automatically suggest an answer for a question given a database off possible previous occurrences of the question given to others is a focus of many studies in many fields.

The work of Roger Schank2, is widely held to be the origin of CBR. He proposed a view on model-based reasoning inspired by human reasoning and memory organization: If a “memory packets” contains a situation where a problem was successfully solved and the person finds himself in a similar situation, the previous experience is recollected and the person can try to follow the same steps in order to reach a solution. Thus, rather than following a general set of rules, reapplying previously successful solution schemes in a new but similar context solves the newly encountered problems.

In the problem solving algorithm of case based reasoning the following steps are taken:

* Describing the current problem.
* Searching for a similar previously solved problem.
* Retrieving the solution to it.
* Adapting the solution to the current problem.
* Verifying the solution.
* Storing the newly solved problem.
* In turn, since the newly found solution may be used for solving future problems, the process denotes the CBR working cycle.

3. Machine learning:

Machine Learning is the wish to program computers so that they can “learn” from input available to them1. The input to our learning algorithm is training data from our company’s database in which stored all technical issues that was dealt over the years, and the correspondences are representing experience which can the program learn in order to advise to a novice support engineer.

To find the best way to do this we’ll have to be more explicit about what data our programs will access, how they parse it, how and when the learning process will be done and will it be evaluated3.

The most popular formal approaches to machine learning applicable to our project are as follows:

3.1 Supervised learning

Algorithms are trained using labeled examples, such as an input where the desired output is known. In our project, for example, the learning algorithm receives as an input a question by the customer and the reply from the customer to this answer by the engineer. The algorithm then learns how good it was based on determining customer’s nature of reply and modifies the model (score and stage) accordingly1,4,5. It will reach this phase if after the “unsupervised learning” phase, directed by the “semi-supervised phase”.

A major sub section of the "Supervised Learning" phase is "Ranking" approach:

* + 1. Ranking

Ranking is the problem of ordering a set of instances according to their “relevance.” A typical application is ordering results of a search engine according to their relevance to the query5.

Our ranking algorithm implementation:

Stores every engineer’s message to customer as possible solution divided into categories with initial ranking of 5 points.

In case two engineer’s messages are “identical” add to current solution one point.

If the case has been closed within:

One more replay from the customer - then add 2 more points to solution.

Three more additional correspondences- then add 1 more points to solution.

Upon receiving customer’s feedback the NLP algorithm identifies it as “positive” feedback or “negative” (alternative: the comment is being marked manually by the support engineer as such)

If “negative” - deducts two point from the solution rating (in case the solution gets below 0 points then it is rejected and cannot be suggested as solution).

If “positive”- add one point to solution rating.

In order to recognize similar solutions suggested in different engineer’s message, we compare each word from the two messages, if 70% of the words in one message were found in the second message then the two can be referred as “identical”.

3.2 Unsupervised learning

It is used against data that has no historical labels. And they have no scoring points for a data set. The goal is to explore the data and find some structure within 4,5 . In our project it is used in the initial phase of the problem solution by looking for the error in the "Mellanox Care Booklet" which is a pre-defined file with common errors and their troubleshooting steps.

The solutions found in this phase will be provided regardless of their ranking in the supervised learning phase.

3.3 Semi-supervised learning

It is used for the same applications as supervised learning. But it uses both labeled and unlabeled data for training – typically a small amount of labeled data with a large amount of unlabeled data .This type of learning can be used with methods such as classification, regression and prediction4,5.

In our project, the semi-supervised approach will be used in two steps:

The Category info found in the Booklet will be used to classify and to provide a prediction of the error in the "supervised learning" phase.

If a solution is not found neither in the “supervised and unsupervised learning" phase. The Date analyzer will suggest a pre-defined solution - to escalate this issue to second tier support engineering team.

3.4 Reinforcement learning

It is often used for robotics, gaming and navigation. With reinforcement learning, the algorithm discovers through trial and error which actions yield the greatest reward. It is known as a “reward hunting” because each time an “agent” performs an action in its environment, a reward or penalty to indicate the desirability of the resulting state. The goal of the agent is to learn an action policy that maximizes the total reward it will receive from any starting state1,4,5. In our project, the desirable goal, is to find solution to the problem the costumer is facing in the shortest amount of replies, minimizing the time effort and cost by the customer. The faster the solution – the bigger the reward. If the decision making process reaches the goal much faster than the fastest solution so far, the score of this solutions is modified, it means a faster solution gets a bigger score.

3.5 Decision Trees

It is one of the most widely used and practical methods for inductive inference. It is a method for approximation of discrete-valued functions, in which a tree represents the learned function. Each node in the decision tree specifies a test of some attribute of the query instance, and each branch descending from that node corresponds to one of the possible values for this attribute 3,5 . In our project, a positive or negative answer by the customer to any of the engineer’s questions in the previous stage (that was preferably generated by this app as well to follow this tree) will eliminate many other troubleshooting steps and may focus on other, more specific, test to find the root cause of the error.

1. Formal definition of AI Implantation

Sensors:

The support engineer.

Actuators:

Suggest a multi stage work-flow based on high scoring answers for every stage and determining what high scoring directions should not be implemented, due to prior actions (fault isolation).

Percept:

Booklet, comments from cases, case resolution, snapshot, switch system logs

4.1 Procedure - Algorithm and flow of events

4.1.1 Artificial intelligence “Mellanox care event” handling:

* An error is being reported by the Mellanox care system
* Salesforce “case” is automatically opened for this error, the case includes: timestamp, device name, source, failure description.
* The support engineers attaches the case logs or Daily Report logs into the Data-Analyzer app.
* The Data-Analyzer compares the source and failure description with previous open Mellanox care cases.
* If the error does not appear in the opened case or in known issues, then the issue needs to be addressed.
* The Data-Analyzer classifies the error type according to the booklet: software, hardware, and configuration. The class can be combination of types (6 types in total)
* If error cannot be found in the booklet it will be classified as a general error (can be the 3 types) therefore the solution for all types will be checked (Unsupervised ML).
* The Data-Analyzer receives the relevant data from the case and determines the case priority: P1 for fatal. P2 for medium. P3 for low priority.
* The Data-Analyzer describes the error in detail for the support engineer: class, potential root cause, more detailed description, overall actions to resolve issue. (If error is found)
* The Data-Analyzer searches within Salesforce for solutions and ranks them as we stated above.
* The Data-Analyzer searches within the category’s solution for the highest ranking solution.
* A solution can be skipped by the support engineer.
* The error type decision branch can be changed.
* If solution bank is empty, the appropriate message will be sent and the issue will be escalated to second support tier to handle.
* If unclassified solution is regularly being resolved by a certain type solution a message to the engineer will advise to manually classify this error.
* Upon receiving the desired suggested solution the support engineer provide it to customer.
* In case the customer feedback is “negative” (this is decided by NLP algorithm, or manually by the support engineer), the support engineer can run the skip solution until solution bank is empty.
* The answer suggestion process is being activated again in a loop (for both negative and positive feedbacks), closing in on the root cause of the failure answer ranking and decision trees, and moving on with customer’s replies.
* The root cause had been found, the issue is resolved.
* Case can be closed.

4.1.2 Artificial intelligence “Daily Report” handling:

* Salesforce case is automatically received in a pre-determined time for the daily report.
* The support engineers attaches the case logs or Daily report logs into the Data-Analyzer app.
* The Data-Analyzer detects the errors from daily report.
* The Data-Analyzer compares the new daily report with yesterday’s daily report.
* If the error appears in the open case, a message is displayed that the issue has not been resolved the issue needs to be addressed again, beginning with step #3 in “Artificial intelligence Error handling”.
* If the above, then the support engineers closes the opened case and adds a “positive” feedback that the issue was solved.
* If the detected error is listed in known issues then need to ignore this issue.
* If not, the Data-Analyzer displays all new errors.
* For all new errors, cases need to be opened.
* The support engineer will open a case manually and proceed with step #3 in “Artificial intelligence Error handling”.

4.1.3 Artificial intelligence “Statistics”:

* The support engineers attaches the problem description into the Data-Analyzer.
* The Data-Analyzer gathers: timestamp, device name, source, failure description for statistical calculations.
* The Data-Analyzer searches in SF and shows how many cases recently opened for this error, this source, and this company.
* The Data-Analyzer describes the error in detail for the support engineer: class, potential root cause, more detailed description, overall actions to resolve issue. (If error is found).
* If unusual grouping of errors is found, the Data-Analyzer calculates the threshold breaches on the groups (threshold adaptation advisory).
* On request, the Data-Analyzer provides statistical data on monthly, weekly and daily scale on case opened and alerts on unusual grouping of errors.

Notes:

The solutions bank will be built in a tree shaped DB. Which consist of 6 branches for all combinations of errors classes: HW, SW, configuration, HW+SW, HW+ configuration, SW+ configuration, HW+SW+ configuration (general or uncategorized). To provide better fault isolation no hop between branches will be allowed (except for override) – the next best solution must be from this decision tree and not a high ranking solution for an unrelated error.

4.2 Determining a “positive” or “negative” response

Determining a “positive” or “negative” response by the customer has a vital role in scoring a solution. Positive scoring means the given solution is helpful, thus it should be considered higher in the order to be reused in further cases with this kind of error. The other way around for the negative response from the customer.

Currently we are considering 2 ways to implement the classification of the nature of the response: manual and NLP based.

4.2.1 Manual classification:

Upon receiving a reply from the customer replying to the solution proposed by the support representative, the support representative will accordingly mark the proposed solution leading to this reply as positive or negative based on the reply.

The marking will be done via two checkboxes: one for positive and one for negative.

A solution can stay in a “natural” state when the classification is not determined.

4.2.2 Natural Language Processing (NLP)

The idea of computers being able to understand ordinary languages human is being a focal point of scientific studies for many years7, in our project we deal with customers from all around the world so when taking in consideration that people from around the world communicate in various ways, and have different ways of expressing themselves and not only do individuals but also people from other countries tend to make choices may be more characteristic. When add to that is the challenge of “English as a second language”, we will fully appreciate the challenges posed by understanding of natural language and even achievements from focusing on a range of specific fields which will be determining what is a positive or a negative response by a customer.

Tokenised text and pattern matching:

One of the more basic operations that can be applied to a text is tokenising: breaking up a stream of characters into words, punctuation marks, numbers and other discrete items 6,7 a fair amount of information may be obtained from relatively shallow analysis of tokenised text for example negative words like “no” didn’t help” may reveal that the reply by the engineer wasn’t helpful while “expressions like “issue resolved” “thanks for the help” may indicate the it was helpful. In addition, we need to pay attention to some aspects that may reveal that the customer is not satisfied such as the text is written with all capital letters or there are numerous or more than two consecutive exclamation marks. The emotional aspects of the language may vary from language to language.

Parts of speech and structure:

A further stage in analyzing text is to associate every token with a grammatical category or part of speech (POS) 6,7. In our project we will have to analyses the grammar of the sentence to determine on the type of the response, such as: “we solved the issue”, “the issue was solved” “your solution was helpful” etc.

1. Design (GUI, UML diagrams)

5.1 Connection Diagram:

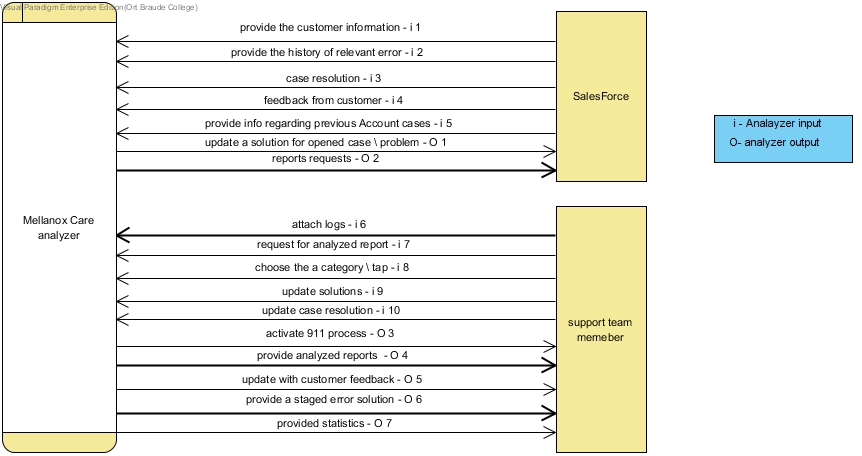


Fig .8 Connection Diagram

5.2 Activity Diagrams:

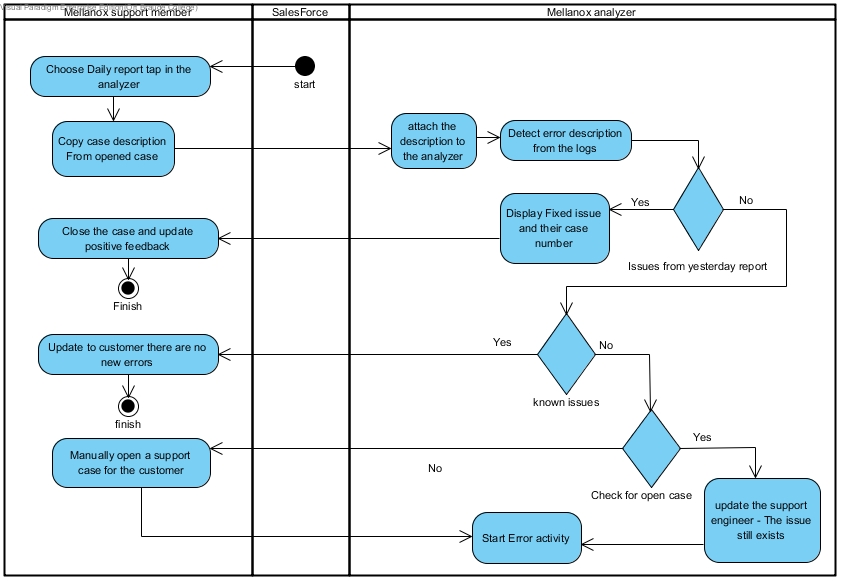


Fig. 1 Activity- Mellanox Care Daily report

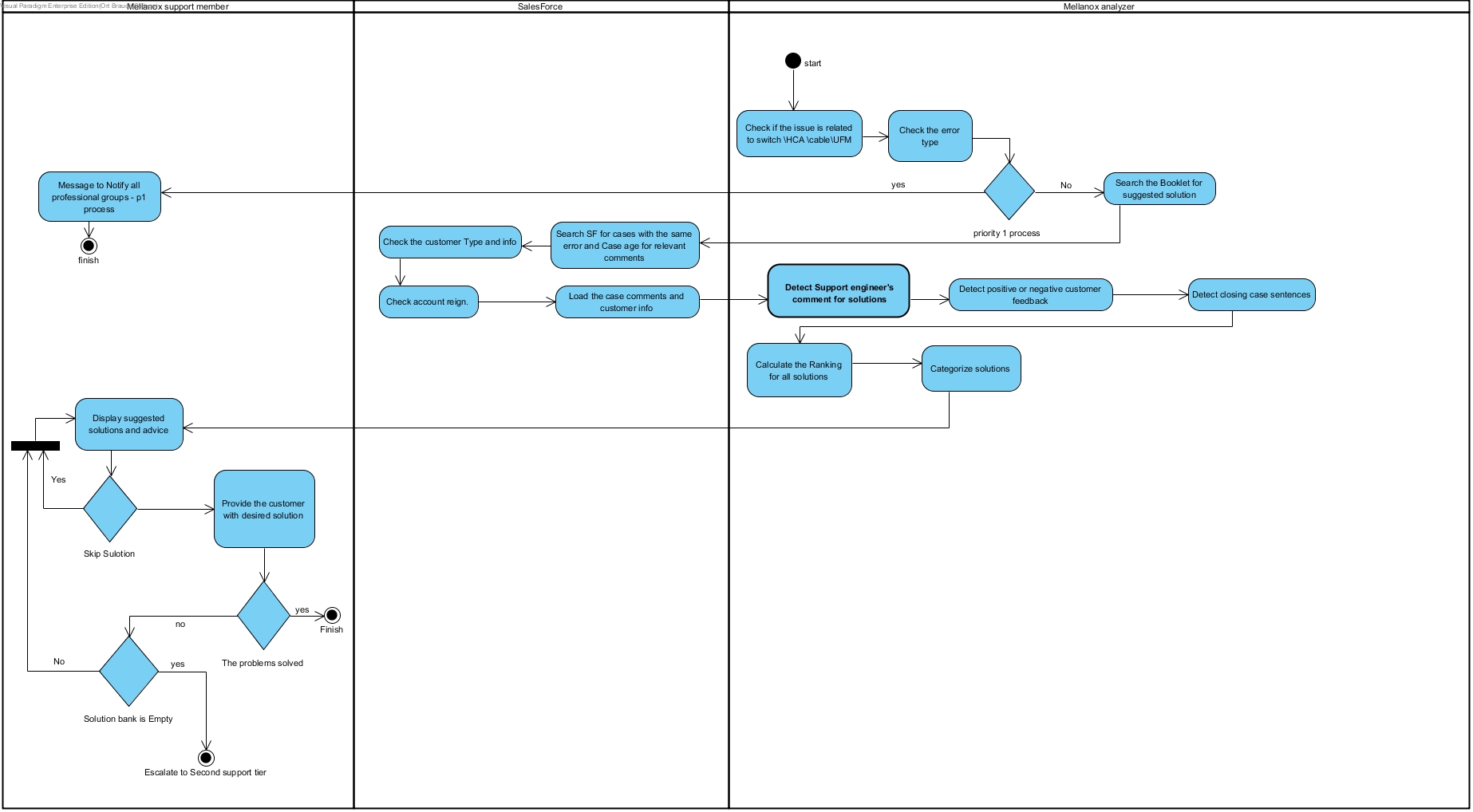
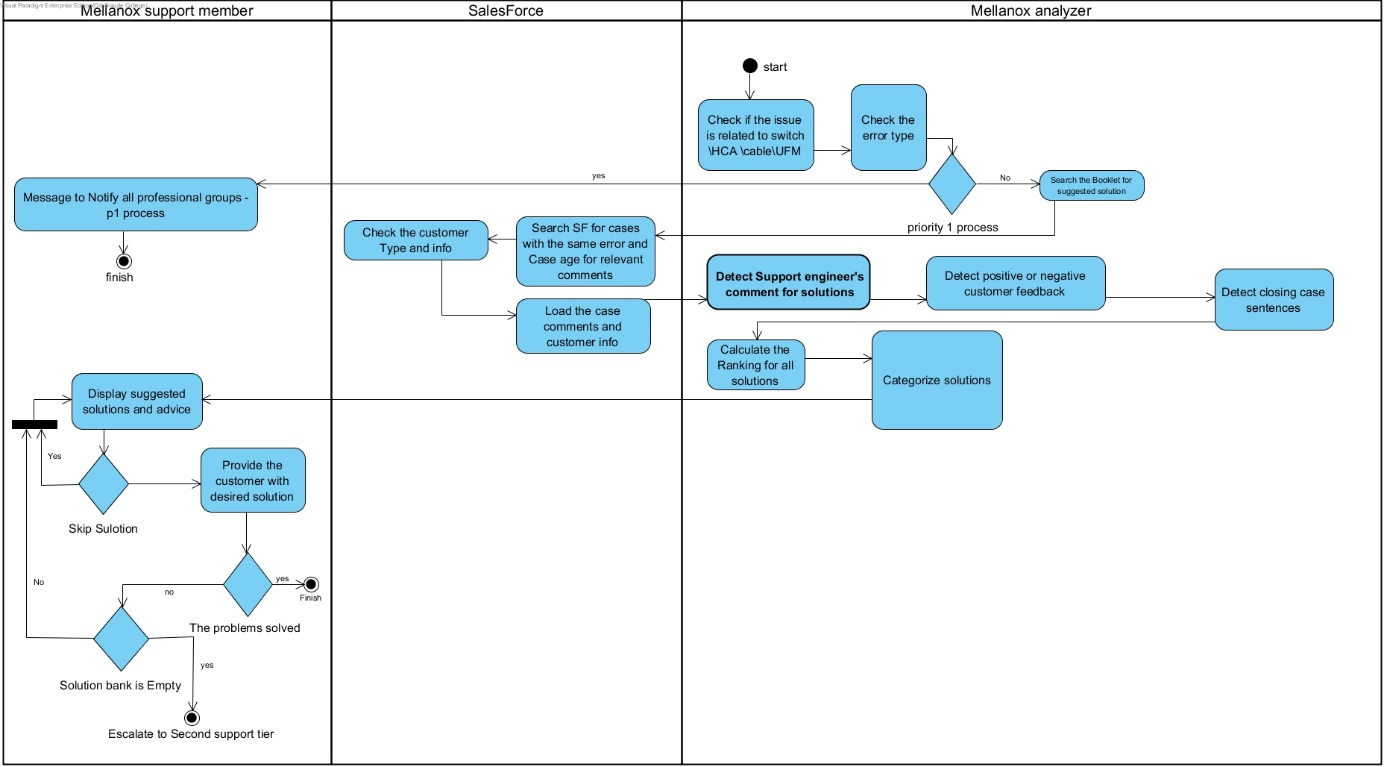


Fig .2 Activity - Handle Error

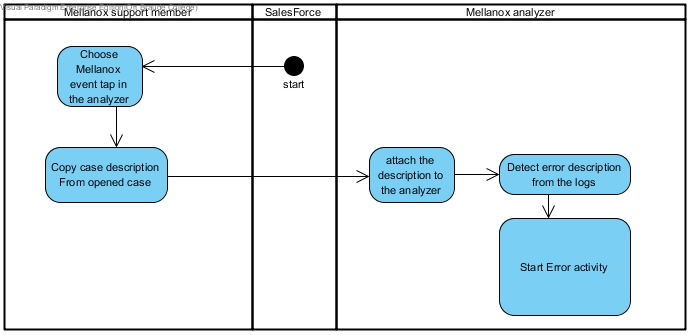


Fig .3 Activity - Mellanox Care Keep alive event:

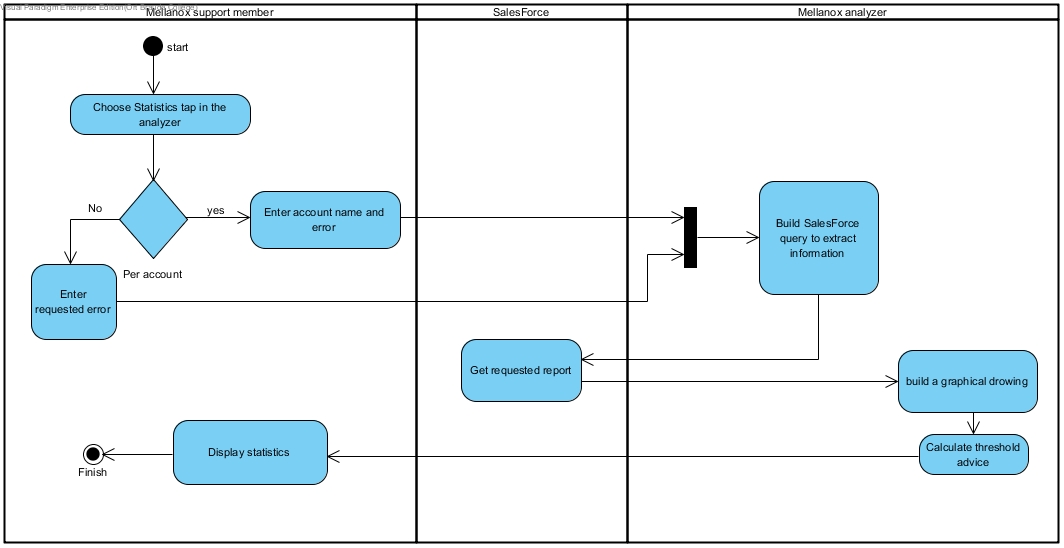


Fig .4 Activity- Show statistics

5.3 Class Diagram:

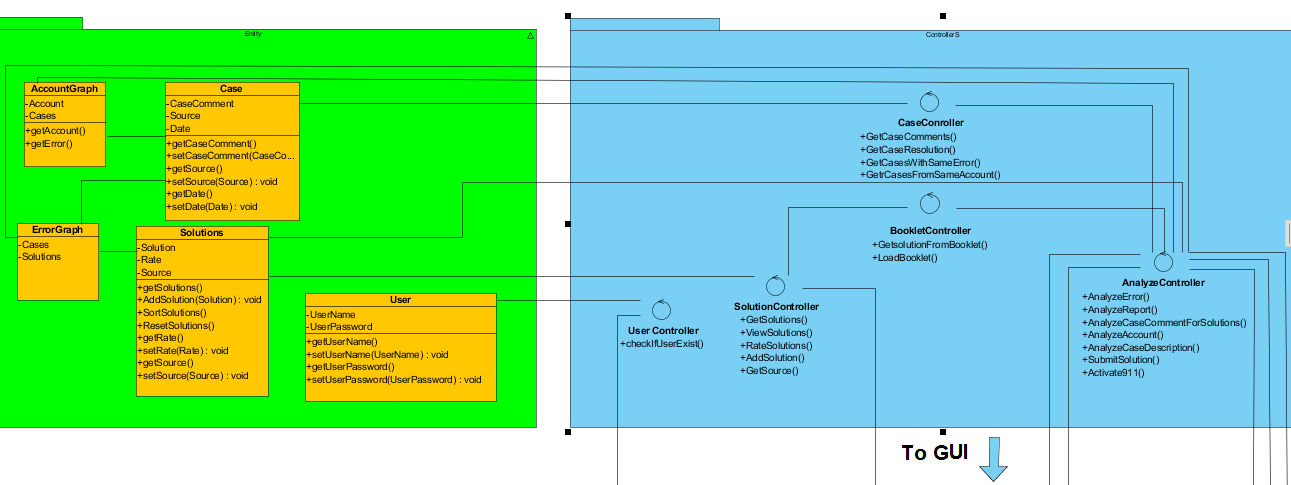


Fig .5 Class Diagram part 1

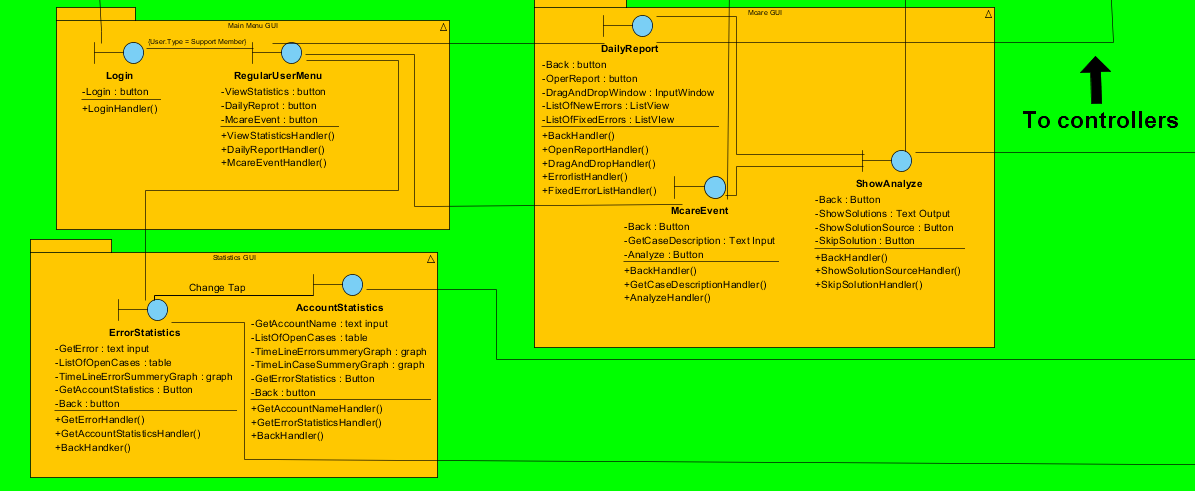


Fig .6 Class Diagram part 2

5.4 Data Flow Diagram:

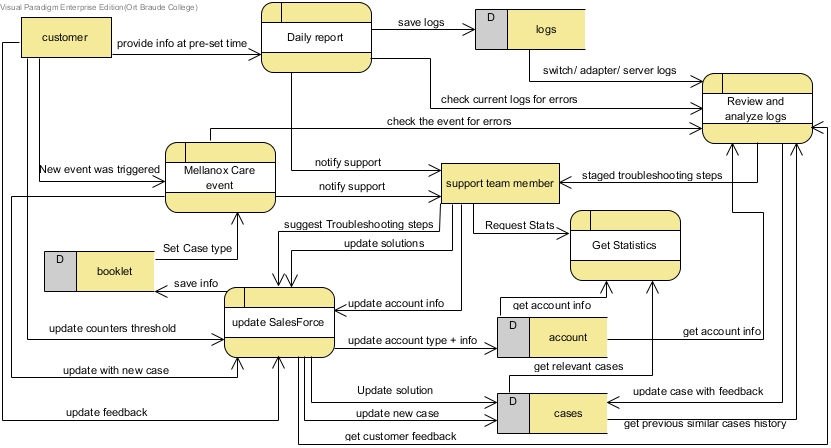


Fig .7 Data Flow Diagram

1. Graphical User Interface (GUI) :

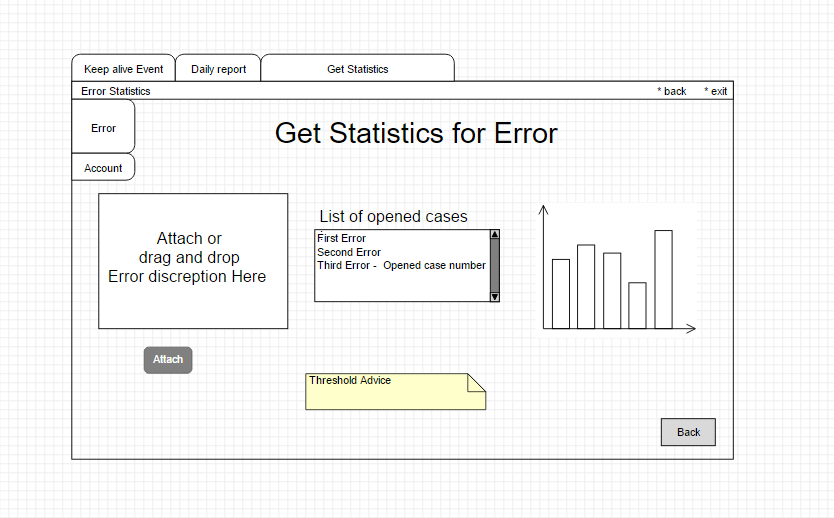


Fig .9 Get statistics for error GUI



Fig .10 Mellanox care Keep alive GUI

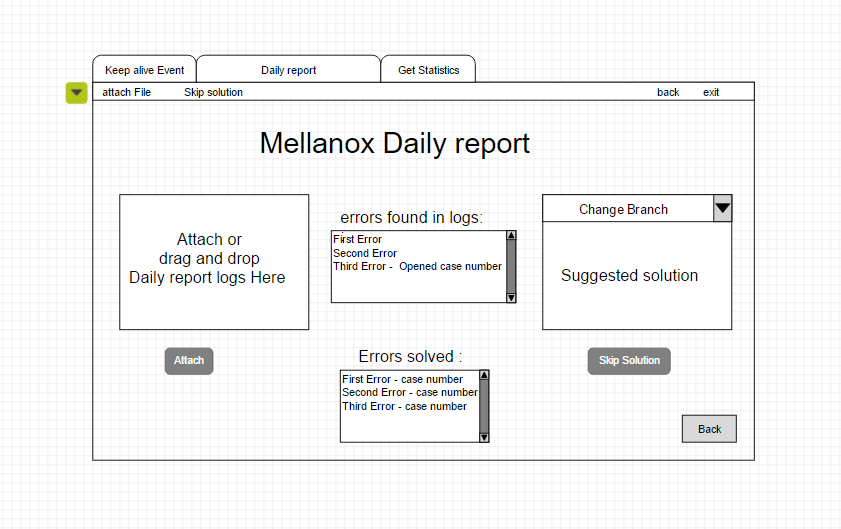


Fig .11 Mellanox care Daily Report GUI

1. Testing plan

We will implement our approach in the context of the alarm triage problem submitted by a customer. Our simulation experiments with data from many previous cases in which the troubleshooting steps by the engineers are different and the responses by the customer are different. We will add new engineers replies and new customer responses to show the potential of our constantly-updating machine-learning-based approach for improving the error triage and troubleshooting process.

|  |  |  |
| --- | --- | --- |
| Test No. | Test subject | Expected result |
| 1. | Attach Mellanox Daily report with only known issues and issues with opened cases. | To show a message that the report is clear with all numbers of open cases related to the detected issues. |
| 2. | Attach Mellanox Daily report with new error (reported for first time). | To provide the support engineer a message with detailed problem description and a possible solution. |
| 3. | Attach Mellanox care new error log related to hardware issue in order to find a solution. | To provide the support engineer a message with detailed problem description and to suggest to replace the faulty hardware. |
| 4. | 1) Attach Mellanox care new error log related to software issue in order to find a solution and then to skip first solution.  2) Update the customer feedback as positive in Salesforce.  3) Attach Mellanox care event log with the same as previous one. | 1) To provide first and second solution.  2) To provide the second solution as first (has higher score- due to positive feedback). |
| 5. | 1) Attach Mellanox care new error log related to configuration and hardware issue in order to find a solution  2) Negative feedback from customer as a response for first solution. | 1) To provide a configuration guide as a solution.  2) To provide hardware replacement solution |
| 6. | Attach an error log in order to receive statistics. | To show a graph with previously opened cases and threshold advice. |
| 7. | To provide account name in order to receive statistics. | To show a graph with previously opened cases. |
| 8. | To provide account name that doesn’t exist | A message to such an account |
| 9. | Attach Mellanox care new error log for new issue never been handled in order to find a solution. | A message to escalate this issue. |
| 10. | Attach Mellanox care new error log in order to find a solution and skip all suggested solutions. | A message need to escalate this issue. |

1. Bibliography:

[1] M. Pantic, *“Introduction to Machine Learning & Case-Based Reasoning”,* 2011.

[2] R.C. Schank, *“Dynamic memory: A theory of reminding and learning in computers and*

*People “,* Cambridge, UK: Cambridge University, *Press, 1982.*

[3] T.M. Mitchell*, “Machine Learning. Singapore”,* McGraw-Hill Companies Inc, 1997.

[4] O. Chapelle, B. Schölkopf, A. Zien *,“Semi-* *Supervised Learning”, 2006*

[5] *S.S. Shwartz, S. Ben-David, “Understanding Machine Learning From Theory to Algorithms”, 2014*

[6]L. Bottou, R. Collobert, J. Weston, M. Karlen, K. Kavukcuoglu, M. MICHAEL, P. Kuksa, *“Natural Language Processing (Almost) from Scratch“,* journal of machine learning research, august 2011

[7] R. Kibble*, Introduction to natural language processing, 2013*

[8]S. Amershi, B. Lee, A. Kapoor, R. Mahajan, B. Christian, *“Human-Guided Machine Learning for Fast and Accurate Network Alarm Triage“,* International Joint Conference on Artificial Intelligence , 7.2011