A-Plan: Integrating Interactive Visualization with Automated Planning for Cooperative Resource Scheduling

Thomas Schneider
Institute of Software Technology & Interactive Systems
Vienna University of Technology, Austria
e0326062@student.tuwien.ac.at

Wolfgang Aigner
Institute of Software Technology & Interactive Systems
Vienna University of Technology, Austria
aigner@ifs.tuwien.ac.at

ABSTRACT
Assignment of staff to work tasks is a complex problem that involves a large number of factors and requires a lot of expertise. Long term as well as short term requirements need to be met which demands flexible solutions. Software tools can aid planners in reaching optimal dispatching plans but currently available solutions provide only incomplete support. This paper describes the design, development, and evaluation of a prototype for semi-automated assignment planning called A-Plan. We have carried out this work in the context of a gas device maintenance provider. In A-Plan, assignments of service technicians to customers are displayed visually and can be modified by direct manipulation. Smooth cooperative work is possible and an optimization algorithm has been integrated that facilitates semi-automatic planning. A qualitative evaluation with potential users and IT professionals provided encouraging feedback on the proposed integration of automated methods and interactive visual interfaces.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces And Presentation**]: User Interfaces—*Graphical user interfaces*; G.1.6 [**Mathematics of Computing**]: Numerical Analysis—*Optimization*

General Terms
Design

Keywords
Visual Analytics, Resource Scheduling, Optimization

1. INTRODUCTION
Proper resource utilization (e.g., staff, machines, rooms, vehicles) is one of the most pressing cost factors in many economic areas. For example a service provider for gas devices needs to maintain technical equipment within certain maintenance intervals. In addition to those regular maintenances that can be planned well in advance, sudden defects might occur and have to be repaired promptly. This demands adaptive and manipulable scheduling. Apart from that, also different skill sets and levels are needed for certain kinds of maintenances or defects which increases the complexity of the problem. On the one hand, personnel needs to be scheduled according to these constraints in order to keep downtime as low as possible. On the other hand, the amount of needed employees should be kept as low as possible and their utilization should be as optimal as possible. Adding to that, different regulations like for example laws on working time need to be followed. Other examples are airlines that have to maintain their aircraft or mobile nursing care companies that have to dispatch their staff to the patients depending on their condition. Especially in the latter case, it is also very important to minimize the distances between the assignments because the necessary travel between the patients causes costs for the company.

In our work, we collaborated with a gas device (e.g., heaters, stoves) maintenance provider that has to dispatch its service technicians to their customers. Currently, the planners schedule assignments for the technicians manually for the next weeks. A major challenge is to optimize the distances between the customers. They also have to be aware of absences of the technicians and the number of assignments per day is also restricted, depending of the time of the year (e.g., in autumn failures of gas devices are more likely than in summer). Moreover, some of the technicians have special skills for special gas devices. All these constraints have to be taken into consideration by the users and are not supported by the currently used software.

The described setting is an example for an optimization task. More precisely it is a linear programming task that can be described by the following model:

$$Min!/Max!: c_1x_1 + c_2x_2 + c_3x_3 + ... + c_nx_n \qquad (1)$$

with

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + ... + a_{1n}x_n <= b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + ... + a_{2n}x_n <= b_2$$
$$...$$
$$a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + ... + a_{mn}x_n <= b_m$$

The first function is the linear function to be maximized or minimized (in our case we want the minimum distance). The other functions are constraints. $x_1 \ldots x_n$ are the variables to calculate (when calculating a route plan, the variable is 1 when the customer will be visited, and 0 when not), $c_1 \ldots c_n$ are the cost parameters (distances between the customers in our case). Examples for constraints are that the number of assignments per day is limited or that there are certain skills of technicians needed. Another constraint of the model is that $x_1 \ldots x_n$ can only be integer numbers. Optimization problems of this type are called combinatorial problems or integer programming. The specialty of a combinatorial problem is the exponential growth of possibilities for a larger number of variables. There are for example 3,628,800 possibilities to bring 10 objects in sequence, for 20 objects 20! is a number with 19 digits.

The given examples illustrate the relevance of this problem and the need for tool support. Better scheduling can help to reduce costs and increase the quality of service for the customers while reducing the administrative work as well as travel and vacancy times. Particularly, a Visual Analytics (VA) approach that integrates automatic methods and supports humans via interactive visual interfaces [21] seems to be a perfect fit for this kind of problem complexity. In this paper we present the design, implementation, and evaluation of a VA prototype that combines automatic optimization and interactive visual interfaces to aid employees of a gas device maintenance provider to schedule their service technicians. The three main challenges in this context are to (1) design interactive visualization methods; (2) integrate automated planning functionality; and (3) support synchronous collaboration when handling customer requests. Our main aim is to aid personnel in their complex planning and scheduling tasks while keeping an optimal balance between automatic algorithms and user interaction.

First, we will present some related research work in the areas of interactive visualization and optimization as well as some widely-used commercial systems in this field. After that, we will report on our user and task analysis in Section 3 that has been conducted in order to inform and guide the design and implementation of our prototype which will be presented in Sections 4 and 5. To assess the utility and usability of our approach, an evaluation has been conducted as described in Section 6. Finally, we will summarize and discuss our findings in Section 7.

2. RELATED WORK

In the following, we will discuss related work along the lines of Information Visualization (InfoVis), optimization, and commercial scheduling systems. For scheduling tasks, visual representations need to be considered that are able to represent time intervals (rather than time instants). An overview of visualization techniques for time interval data is given by Aigner et al. [1]. Timelines are a simple and widely-used representation of events or time intervals. Karam describes timelines as linear graphical visualization of events over time [15]. In general the time axis is presented along one display dimension (mostly the horizontal axis) and category along the other axis. Events are shown as lines or bars whereas their length represents the temporal extent of an event. Plaisant et al. [18] introduced LifeLines as a further development of timelines. In LifeLines the thickness and color of the bars can be used to encode additional information. Plaisant et al. also included interactive features, e.g., for overview+detail and zooming. GANTT charts are a special form of timelines invented by Henry L. Gantt (1919) which are ubiquitously used in several project management products today. Modern project management software also shows milestones and uses hierarchies of tasks to keep the complexity at a manageable level. The disadvantage of the aforementioned visualization techniques is their high space consumption as inactive intervals need to be represented as blank areas. Therefore, it is difficult to simultaneously investigate many categories on whether they are running in parallel. Luz and Masoodian [17] try to decrease the space consumption in their Temporal Mosaic technique by visually aggregating related interval bars into compound rectangles. Color is used to distinguish between different categories of events. However, moving assignments in temporal mosaics via direct manipulation is difficult which renders them inappropriate for scheduling tasks.

Because of the exponential growth of possibilities for combinatorial problems, only approximation algorithms can be used. For the special problem described in the previous section a large amount of literature can be found. The multiple traveling salesman (M-TSP) is a model where some salesmen have to travel to customers [4]. The goal is to travel to all customers with minimal costs. The generalized formulation of the problem is the vehicle routing problem (VRP). The VRP is an M-TSP with cargo capacity restrictions. Cargo capacity can be interpreted as a restriction for the number of customers that can be visited by a technician per day. Several approaches exist for VRP. Genetic algorithms are adaptive heuristic search methods based on population genetics [5]. A dynamic genetic algorithm can be found in [14] where dynamic means that new customer orders can be dispatched after the planning has completed (in our example a failure assignment). A similar solution with stochastic customers is described in [6]. When developing algorithms it is important that the calculation is fast (nearly in real-time) for a huge number of customers (about 20,000). The result of the algorithm is a suggestion for the plan that can be adapted by the user. Two examples are the sweep algorithm and the savings algorithm. The sweep algorithm [13] assumes that the locations of customers and the headquarter are given by coordinates (x_i, y_i) and the dispatching point is in the origin of the coordinate system. The distances between the locations are determined as Eucleadian distances. The savings algorithm [7] is the most widely known heuristic for the VRP [22]. The idea behind it is that savings can be obtained by joining two routes into one route.

Due to the relevance of resource scheduling problems in industry, also a large number of commercial and open source software products are available. Examples are SAP CRM, Microsoft CRM, Service Ledger[1], ORS Online Resource Scheduler[2], phpScheduleIt [3], Flight Schedule Pro[4] , Schedule Pro[5], and Titanium Schedule[6]. The main focus of SAP CRM and Microsoft CRM is customer relationship management, but they also provide service modules and are widely used in companies. More details on the usage and issues with SAP CRM will be presented in Section 3. An interesting feature

[1]http://www.serviceledger.com
[2]http://ors.sourceforge.net
[3]http://www.php.brickhost.com
[4]http://www.flightschedulepro.com
[5]http://www.invisionwfm.com
[6]http://www.titaniumschedule.com
(all URLs accessed at January 20, 2011)

of Service Ledger is its MapPoint integration that shows routes on a map with the approximate travel times. Online Resource Scheduler and phpScheduleIt are open source web applications that offer planning functionality for any resource. Flight Schedule Pro is a specialized software for scheduling of aircraft for flight schools, universities, and flying clubs. For the visualization of the scheduling data, timelines and calendars are used in the mentioned products. The multiple views of Service Ledger provide a calendar overview and details when selecting an entry. Flight Schedule Pro delivers more information when pointing on an entry via tooltips. Schedule Pro distributes all changes to the clients, so all clients always have the actual data which is necessary for cooperative work. Titanium Schedule uses many colors and icons on the scheduling plan to inform the user about the different assignments. Due to this fact the representation is rather cluttered and confusing. Schedule Pro uses a plan representation where only one day is shown in full detail and the other days are shown in a compressed form similar to DateLens [3].

After discussing related work we will now present the user and task analysis we conducted.

3. USER & TASK ANALYSIS

At the beginning of our project qualitative research was conducted to analyze current work practices in order to understand behaviors and attitudes of users as well as technical, business, and environmental contexts (the domain) of the tool to be designed. Furthermore, vocabulary and other social aspects regarding how existing products are used are important to understand the domain in question.

3.1 Method

Kulyk et al. [16] present several methods for user and task analysis: *contextual observation* is a method where the designer observes the user's working environment in practice. Observation is very useful but also has to deal with several problems. The observations can be misinterpreted, e.g., when the observer does not know the context of the actions. An observation can also disturb the work of the observed group and so the observed work can differ from normal work situations. In *interviews* subjects will be asked about their work and the use of the software and artifacts. Bartlett and Toms [2] notice that *"a drawback to the interview approach is that it relies on recall, rather than directly capturing the activity of interest, and is thus vulnerable to missing details that were either forgotten, or not considered relevant." Task demonstration* is similar to observation but in contrast the observer may ask questions and the demonstrator may explain some tasks in more detail. A disadvantage is that the task is described by the user, so the feedback may be very limited and problems may not become visible, since most experienced users are not aware of these problems [20]. To mitigate the disadvantages of the methods we used a combination of task demonstration, contextual observation, and interviews for the analysis. We started with a task demonstration to get the big picture about the currently used application (SAP CRM) and the process. In the contextual observation we found out how the work is really done. Finally, interviews were a possibility to get more information about the users and the problems with the current application. These contained questions like: What experience with IT systems do the people have? What software do they use for their work? What should an optimal system for the desired tasks look like?

3.2 Results

To illustrate the problem, all tasks and currently used software were investigated. In the analyzed company 35 technicians do the service work and seven employees do the planning and dispatching in the office. Three main use cases were identified:

- *Regular assignment for customers with maintenance contract*: All customer with a maintenance contract periodically receive a service assignment for their gas device. This assignments have to be planned for service technicians whereas the distances between the customers should be as short as possible.
- *Failure of gas devices*: Customers announce failures of their gas devices to the call center of the company. The agent tries to fix the problem on the telephone; if the problem can't be solved, she makes an entry in the failure list. Later she tries to contact a service technician near the customer to solve the problem.
- *Refusal or deferral of assignments for maintenance*: When a customer wishes to cancel or change the date of her assignment she also calls in. The dispatcher writes the corrected date of the customer into the printed assignment plan.

When analyzing the current work practice, we found many usability issues in the used software system (SAP CRM). This includes for example that information about the assignments is abbreviated in an unreadable way; the behavior for drag-and-drop of assignments is not consistent (for moving of assignments direct manipulation is used whereas for the insertion of new assignments this is not possible); the usage of colors is confusing because contrary to expectation, color does not describe the current status of an assignment; withdrawing of operations is not possible in the whole system because undo does not exist in the entire application. Moreover, the user interface is overloaded and cluttered, some information is redundant, and the important information is hard to find. Finally, the user has to handle three different applications for conducting the aforementioned use cases.

During contextual observation we encountered that in many cases the dispatcher does not use the software at all. She has to answer questions and confirm dates of customers on the phone within seconds which is not possible with the current system (it takes for example about four minutes to generate a new assignment). Therefore, the users came up with a workaround that makes rapid answering on the phone possible. Rather than retrieving information from the electronic system, two paper-based artifacts are used:

- *A book calendar* that includes all appointment requests from customers. Thus, the customers' data can quickly be entered during the phone call. In the evening all data is recorded in the system, thus the long waiting times of the electronic system to read the data will occur only once for all assignments of the day.

- *A folder* with separator sheets for each technician and a list of assignments per day per technician. The lists are always printed from the SAP system some days in advance. When a customer calls the dispatcher and reports short-notice cancellations or schedule changes, the information will be recorded in this folder on paper.

3.3 Personas & Scenarios

The goal for this user & task analysis besides gaining more knowledge about the domain, user requirements, and desires, was the creation of representative user profiles, their goals, and the construction of interaction scenarios based on this user model. Following the user-centered design approach by Cooper [9], this lead to the creation of scenarios and personas that aid design and evaluation. Personas are a created cast of characters representing real persons along with both their knowledge in the computer and the application domain. These persons have certain goals they want to achieve when using a product. A scenario is basically a detailed story about a person performing a certain task to achieve her goals. In our case, we identified two personas (Erich Gruber, a 50-year-old technician and Julia Steiner, a 30-year-old more business-oriented dispatcher) and we created four scenarios that capture the main use cases: *Appointment Request with Assignment*, *Appointment Request without Assignment*, *Cancellation*, and *Failure*. These personas and scenarios mostly cover the different kinds of employees and use cases in the company.

Based on the results of the user & task analysis as well as the created personas & scenarios, we designed a prototype that will be described next.

4. DESIGN

Two guiding lines of the design of A-Plan were to fulfill user requirements and to avoid reported problems and issues of current work practice. To support and ease the workflow of users, an automated planning function should be integrated into the software. Furthermore, multiple users should be able to work simultaneously with the data while being aware of each others' actions. Following Cooper et al.'s recommendation [10] we eliminated save buttons and avoided OK buttons. Instead, every action should be saved automatically and an undo function should be available for the user to take back unwanted operations. Following that, every transaction should be saved immediately and distributed to the other clients.

Figure 1(a) shows the basic screen layout of A-Plan. The screen is divided into three areas: (1) The *planning area* is the place where the user can view, insert, move and delete assignments. (2) *Details* are shown on the right side of the window. In this location the user can also start actions like searching for a customer and planning of assignments. (3) *Collaboration:* In the lower area of the window the user is supported in the cooperation with other users.

4.1 Planning Area: Visualization & Interaction Design

We decided to use traditional timelines for the visualization of the assignment plan because of its widespread use and ability to display the data characteristics at hand. Furthermore, timelines are well-suited to be used and manipulated interactively and cooperatively by more than one user. In our case, assignments are displayed as boxes showing the most important information about the assignment directly as text (city, customer, time). The timelines are arranged in horizontal lanes whereas a single lane corresponds to a specific service technician. The shape and the color of the box also gives information about the assignment: When the box has rounded corners, only the date of the assignment is fixed but not the exact time within the day whereas when the corners are angular the time is also fixed. The color of the assignment gives quick information about the type of the assignment (maintenance or failure). When hovering over an assignment a tooltip shows more details and when clicking on an assignment all information about the customer is shown in the detail area to the right. This resembles an overview first and detail-on-demand approach, where overview and detail information are displayed simultaneously in a distinct presentation space [8].

Different interactions for smooth panning and zooming allow for navigation in time. I.e., zooming can be performed by using the mouse wheel, a slider in the toolbar, or two buttons next to the slider. Panning is done by dragging the background of the plan or by using a navigation element in the toolbar. Furthermore, users might navigate by directly selecting a day of choice using a date chooser widget. Changes to the plan can be performed by direct manipulation. The user might drag-and-drop assignments in the plan. This form of interaction can also be used to insert new assignments. In this case the user can drag a surrogate assignment from the detail area into the planning area. All changes are distributed to all clients immediately. While one user is dragging an assignment all other users can follow the movement of the assignment live. Until the assignment is dropped the rectangle is rendered transparent on the plan (see Fig. 1(b)). This movement is also distributed to all other clients, so all users can see the movement of the assignment. On the other clients the assignment is also rendered transparent until the assignment is dropped.

The application provides unlimited undo/redo functionality. All changes on assignments are saved in a database and can be made undone by using the undo button. The undo/redo function is user-specific which means that change histories are stored separately for each user.

4.2 Detail Area

The detail area not only shows the details of selected items in the planning area but the user can perform a wide range of interactions: search for customers, insert new assignments, insert new open failures, and plan customers with due maintenance contracts. It is organized using three tabs: The assignment tab, the planning tab, and the open failures tab.

Assignment Tab. The assignment tab (see Fig. 1(c)) provides an interface for four activities: search, detail information, edit, and insert. Customers can be searched by entering data about the customers in the search box. The tool performs a full text search over all data fields of the customers. For example a search for 'Schneider Eisenstadt' will provide all customers with 'Schneider' in the name and liv-

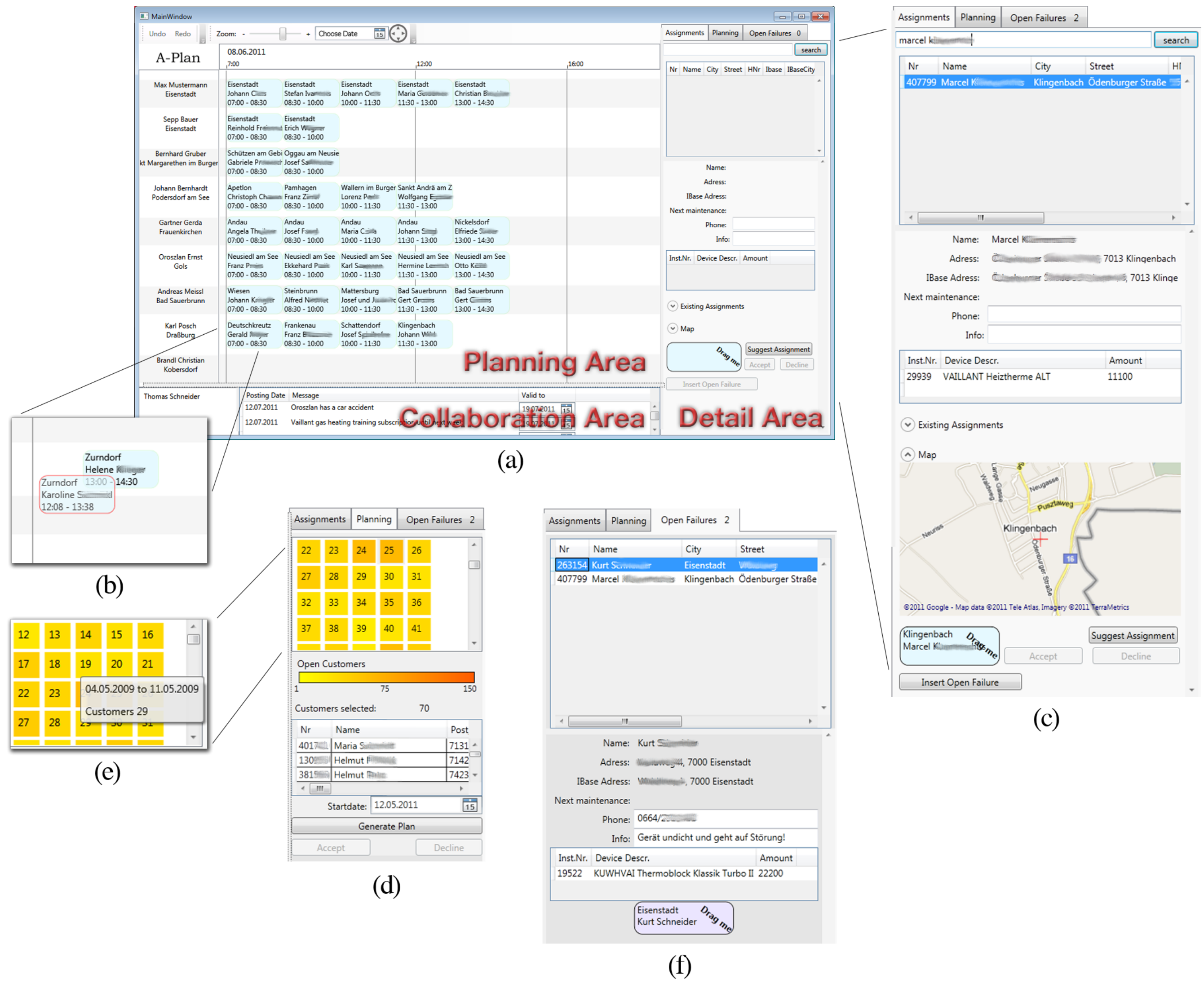

Figure 1: A-Plan: (a) basic screen layout (top: toolbar including undo/redo buttons, zoom slider, date chooser; left: planning area with interactive timeline visualization; right: detail area; bottom: collaboration area including user list and messages) (b) assignment is shown semi-transparent while dragging; (c) assignment tab showing details with map view; (d) planning tab; (e) heatmap view; (f) open failures tab.

ing in 'Eisenstadt', furthermore the result will also contain customers with 'Schneider' in the name and living in the street 'Eisenstadt street' for example. This way to search is similar to internet search engines, which is familiar to most users. This is an advantage over conventional search functionality where the user has to specify the search criteria for each database field explicitly. When a single customer is selected in the search result, the newest existing assignment is shown in the planning area, so the user does not have to pan — the application does this automatically.

Below the search panel, all detail information about a selected assignment is shown including customer data, information about installed devices, and existing assignments. Assignment data and customer data like the phone number can be edited in place. Below that, a map view is integrated, where the location of the customer will be displayed. To insert an assignment the user has two alternatives: drag-and-drop the surrogate assignment below the map to the plan manually or let the system suggest a date for an assignment using an automated algorithm. Furthermore, open failures can be inserted using a button which are then displayed in the 'Open Failures' tab (see Fig. 1(f)).

Planning Tab. The planning tab (see Fig. 1(d)) offers the functionality to do automatic planning of due contracts. The number of open customers are visualized using a heatmap (see Fig. 1(e)) where each square shows the data of one week, the number of customers is shown via color intensity, and the week number is displayed as text. The optimization procedure is initiated by the user upon selection of a number of weeks to plan. For the selected customers a plan will be generated using the savings algorithm [7], as this algorithm is used in many applications in practice [11] and delivered good results in our own tests. The algorithm optimizes the distances between the customers. In the first step, a route plan from the existing assignments is generated. Afterwards, the algorithm creates pending tours between each customer and the technician with the minimal distance to the customer. In the last step, savings by joining two tours are calculated: the algorithm starts with the biggest saving and joins two tours into one under compliance with the restriction (maximum number of tours per day per technician) until no tours can be joined anymore.

Upon completion of the computation, the assignments proposed by the automatic algorithm are shown in the planning area as semi-transparent assignments. The user can now review the suggested assignments and accept or deny the proposed plan. Using this human-in-the-loop approach, automatic planning and human judgment are closely coupled using interactive visual interfaces. If scheduled appointments need to be changed or cancelled, no automatic re-planning is performed. This is necessary because customers already got notified about their appointments and further changes might only be made upon human intervention.

Open Failures Tab. A list of open failures (see Fig. 1(f)) is displayed on the third tab. This list can be seen as to-do list and the dispatcher has to search for suitable technicians and assign them to the open failures. The layout of this tab is very similar to the assignments tab, with the only difference that open failure assignments cannot be inserted automatically: the call center agent has to find a free technician by calling them on the phone.

4.3 Collaboration area

A-Plan supports synchronous and asynchronous collaboration as well as task-oriented and social awareness as suggested by Prinz [19]. In the collaboration area (see Fig. 1(a), bottom) two lists are shown: the active users (left; for synchronous collaboration) and social awareness and common messages (right; for asynchronous collaboration). To facilitate the awareness of presence, all users currently working on the system are shown in the active users list. As soon as they close the application they disappear on all clients. The list of common messages can be used to notify the other users about important news as for example recalls of a gas device provider or vacations of technicians. Apart from that, activities like editing assignments are synchronized across clients and users for seamless coordination between dispatchers.

Overall, three different interactive visualizations are used in A-Plan that are tightly integrated: the planning area using timelines, the map view showing the location of selected assignments, and the heatmap to display amount of open assignments to plan on a weekly granularity. Moreover, an optimization algorithm is used for supporting the semi-automatic planning of assignments. Other more or less standard UI elements frame these core components and form a coherent tool design.

5. IMPLEMENTATION

As proof-of-concept for the described design, we implemented a prototype using C# in Visual Studio 2010 with .net 4.0. For the user interface parts, the WPF[7] framework was used and the communication between the server and the clients is implemented with WCF[8]. In the implementation the MVVM (Model-View-Viewmodel) pattern was used. The goal of MVVM is to keep the code as maintainable as possible by separating the user interface from the logic as strong as possible. The freely available MVVM light toolkit[9] was used to facilitate the work with the MVVM pattern.

From an architectural point of view, we followed a client-server approach for coordination. We used the net.tcp binding of WCF which is restricted to WCF applications only, but offers better performance. The interoperability provided by other bindings was not necessary for our application. In the application all calls between clients and the server are made asynchronously. If the call would be synchronously the processing of the application would be suspended until the call of the service is terminated. As these waiting times would cause interruptions for users, we decided to develop the communication between server and client with asynchronous calls. At the end of each function on the server the client is called with the result as parameter.

6. EVALUATION

In order to assess the usability and utility of our design and implemented prototype, a qualitative evaluation with domain experts was conducted. The goal of this evaluation was less focused on validating the correct behavior of the prototype, but on investigating to what extent A-Plan meets the presented requirements.

6.1 Method & Participants

Our evaluation was structured into three parts: First, the prototype was demonstrated to the participants explaining the interface and basic functionality. After that, user testing took place where subjects had to carry out a set of given tasks. During that the thinking-aloud method was applied [23], i.e., participants were encouraged to verbalize their thoughts. Finally, semi-structured interviews were conducted in order to reflect on the design and usability of the prototype as well as gathering input on perceived advantages and disadvantages over the current work practice and suggestions for future work.

Five persons participated in the evaluation of A-Plan. Three of them were female and two male with an age between 26 and 49 years. Four out of them were domain experts with 2, 4, 15, and 20 years of experience in that area. Two of the domain experts work in customer care, one is a customer service technician, and one is an IT expert. One of the subjects was a master student in computer science who was not familiar with the domain and had no experience with planning software. The main focus was on the potential users of the system (the customer care agents). The reason to test the prototype with a service technician is that in times where the work load is very high for customer care agents, a service technician has to help in the back office. With the two IT experts we wanted to get more critical feedback about the usability of the prototype.

[7]WPF = windows presentation foundation

[8]WCF = windows communication foundation

[9]http://www.galasoft.ch/mvvm/getstarted (April 24, 2011)

6.2 Material & Analysis Approach

For the test a PC and a laptop were used to demonstrate the possibilities of collaboration. During this process we made audio recordings and the activities of the system were recorded by a screen recorder software. In addition, written notes were taken by the study facilitator recording the activities of the testers. Through this multiple logging approach we wanted to avoid that interesting aspects will not be included in the analysis of the evaluation.

For the user testing, users had to carry out seven different tasks. These tasks were developed in order to cover the most important use cases identified in the user & task analysis as well as evaluating the novel interactive features and automatic planning functionality. Examples for the posed tasks are "The customer 'Kurt Schn..., Eisenstadt' reports a device failure, record this case", or "Move this assignment to the next day and set the assignment as fixed".

In the semi-structured interviews questions about the application, about the visualization of the plan, about the detail area (assignments and open failures), and about the long term planning functionality of A-Plan were used in the interview guideline.

The written notes and audio recordings of both the user testing and interview phases were analyzed along the proposed categories by Forsell & Johansson [12]. These are heuristics specifically developed for evaluating information visualizations to assess common and important usability problems.

6.3 Results & Discussion

The general feedback of study participants was very positive and a number of shortcomings and future improvements could be identified. When analyzing the found problems and issues based on the heuristics of Forsell & Johansson [12], most of them were of the categories *B7 Orientation and help* and *B5 Information Coding* (both 10 times), followed by *E7 Minimal actions* (8 times). In the following we will provide more details on the gathered data.

Three persons (the customer care agents and the service technician) noted the speed in which the required tasks can be solved with A-Plan. For example the scheduling of a customer for maintenance in the current system can be estimated with four minutes effort, in A-Plan this can be done within a few seconds. The search function received much praise as well. As the way to search in A-Plan is familiar to all users from search engines it was conceived as being a very easy and fast way to find the desired data. The overall screen layout was clear to all testers. For three testers it was a little overloaded in some areas, but they also mentioned that they have no concrete suggestions for improvements. For one customer care agent the simple way to modify the data was unfamiliar, she would prefer an additional confirmation action for some operations. This might be attributed to the fact that this is the way how to work in the current system. One tester would prefer to use a save button instead of the automatic save with undo/redo capability. The program crashed in some situations. This was not a problem for the testers, as we educated them about the early development and test state of the system but for a productive system bugs should be corrected.

The visualization of the plan using timelines was clear for all testers. The functionality to show details by tooltips has not been recognized by the users. Only after we gave them a hint, they used it but found the function useful. The way to move assignments by drag-and-drop was no problem for the testers. The way to get details of assignments earned positive feedback too. However, of course also several problems were identified, some of which occurred for several or even all of the participants: The most common problem for the users during the prototype evaluation occurred when shifting assignments. The task was to move an assignment to the next day and all users tried to do this by dragging the timeline of the assignment to the next day. The problem occurred when they reached the border of the view which would not pan automatically and they could not move the assignment anymore. For the test persons the distinction between fixed and not fixed assignments using shape only was not strong enough. They would prefer to use different colors or a border around the fixed assignments. For three testers the way to zoom and pan was very unfamiliar, they would prefer a fixed view where no free panning and zooming is possible. One test person mentioned that as the movement of an assignment by drag-and-drop can happen unintentionally, she would prefer to move assignments with the right mouse button. Another problem occurred two times at the insertion of an open failure: two testers clicked the button to insert an new open failure more than once, which created two entries in the list that is hidden on another tab. Here, better system feedback should be provided making the user aware that an open failure was successfully recorded.

An important functionality that is missing, is the activity history of assignments. For example if an agent deletes an assignment, it is hard to comprehend what has been done, by whom, and why because the data does not exist anymore. This has been planned in the conceptual design but was not implemented due to time constraints.

The semi-automatic planning functionality earned very positive feedback. In the currently used system, planning has to be done completely manually which is a very time consuming operation. The computed plan of A-Plan was perceived as acceptable to the study participants and the handling of the planning process was found to be easy and clear. Only one tester had problems to understand the heatmap representation which shows open maintenances to plan.

7. CONCLUSION & FUTURE WORK

In this work we reported on the design, prototypical implementation, and evaluation of a VA tool for scheduling of technicians for gas device maintenance. In the beginning we investigated the field of work in such a company. It became apparent that the standard software used does not support the needed tasks well. Interestingly, paper-based workarounds have been developed by the employees of the investigated company to mitigate these problems. We were astonished to encounter this sophisticated system of paper-based artifacts to reach a more or less smooth working environment. This provided very valuable insights for our own development.

Based on the information gained via contextual observation and interviews, we designed and implemented a prototype called A-Plan that integrates interactive visualizations with automated planning and supports collaborative work. A-Plan uses an interactive visualization for presenting planned assignments that is based on timelines. A detail view of assignments includes an interactive map view for localizing customers. Furthermore, an automated planning functionality based on the savings algorithm has been integrated which allows for bulk planning of recurring service contracts and is supported by a heatmap visualization. The algorithm accounts for a complex set of constraints like geographic areas and timings and is suggesting an automatically optimized set of assignments. The suggested plan can be reviewed and altered by the user via the interactive visual interface.

We evaluated the implemented prototype with three users of the current system and two IT experts. Testing the prototype with people who do their daily work in this field yielded much interesting feedback. Some issues arose from the fact that the users who did the evaluation were not familiar with techniques like direct manipulation. Overall, we received encouraging feedback and were able to identify shortcomings of the design and functionality of A-Plan. All test subjects would prefer to use A-Plan instead of the existing system.

Some general lessons learned for future developments in this area are that timelines are an easily understandable visual representation and allow for intuitive user interaction. Furthermore, combining automatic and visual methods in a semi-automatic fashion is a well-suited approach for this problem area. Especially, using a visual representation to display the suggestions of the optimization algorithm and make them manipulable was praised by users. Fully automating the process might be doable in theory but in that case users might no longer have the feeling of being in control and might not be able to create a mental model of the inner workings of the system. We believe that human reasoning capabilities add value to the planning process and that the taken approach is superior to both, purely manual and fully automatic methods. From a user's point of view, main challenges are to support synchronous collaboration in real-time and that needed information can be found quickly.

As A-Plan is currently in the prototype stage, a number of directions for future work remain. Apart from fixing a number of software bugs the issues that surfaced in our evaluation should be addressed, as for example the movement of an assignment outside the current view, the visual distinction of fixed and variable assignments, and improvements in direct manipulation. Other than that, introducing a semantic zoom functionality could increase and optimize the displayed information.

The main contribution of our work is that we have demonstrated the successful application of a Visual Analytics approach in the context of resource scheduling. We have shown an effective integration of automatic methods and interactive visualizations based on a user-centric development approach. The integration of the strength of both the human and the computer enables the creation a powerful environment for a set of non-trivial and complex tasks.

Acknowledgments. This work was supported by the Centre for Visual Analytics Science and Technology CVAST (funded by the Austrian Federal Ministry of Economy, Family and Youth in the Laura Bassi Centres of Excellence initiative).

8. REFERENCES

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer, 2011.

[2] J. C. Bartlett and E. G. Toms. Developing a Protocol for Bioinformatics Analysis: An Integrated Information Behavior and Task Analysis Approach. *Am. Soc. Inf. Sci. Technol.*, 56(5):469–482, 2005.

[3] B. B. Bederson, A. Clamage, M. P. Czerwinski, and G. G. Robertson. DateLens: A Fisheye Calendar Interface for PDAs. *ACM TOCHI*, 11(1):90–119, 2004.

[4] T. Bektas. The Multiple Traveling Salesman Problem: An Overview of Formulations and Solution Procedures. *Omega*, 34(3):209–219, 2006.

[5] O. Bräysy. Evolutionary Algorithms for the Vehicle Routing Problem with Time Windows. *Heuristics*, 10(6):587–611, 2004.

[6] R. Cheung, D. Xu, and Y. Guan. A Solution Method for a Two-dispatch Delivery Problem with Stochastic Customers. *Math. Model. Algorithm*, 6:87–107, 2007.

[7] G. Clarke and J. V. Wright. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Oper. Res.*, 12(4):568–581, 1964.

[8] A. Cockburn, A. Karlson, and B. B. Bederson. A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. *ACM Computing Surveys*, 41(1):1–31, 2008.

[9] A. Cooper. *The Inmates Are Running The Asylum*. Sams Publishing, 2004.

[10] A. Cooper, R. Reimann, and D. Cronin. *About Face 3: The Essentials of Interaction Design*. Wiley Publishing, 2007.

[11] W. Domschke. *Logistik, Rundreisen und Touren*. Oldenbourg, 3rd edition, 1990.

[12] C. Forsell and J. Johansson. An Heuristic Set for Evaluation in Information Visualization. In *Proc. of Int. Conf. on Adv. Visual Interfaces (AVI)*, pages 199–206. ACM, 2010.

[13] B. Gillett and L. Miller. A Heuristic Algorithm for the Vehicle-Dispatch Problem. *Oper. Res.*, 22(2):340–349, 1974.

[14] F. Hanshar. Dynamic Vehicle Routing Using Genetic Algorithms. *Appl. Intell.*, 27(1):89–99, 2007.

[15] G. M. Karam. Visualization Using Timelines. *Proc. of Int. Symp. on SW Testing and Analysis (ISSTA '94)*, pages 125–137, 1994.

[16] O. Kulyk, R. Kosara, J. Urquiza, and I. Wassink. Human-Centered Aspects. In A. Kerren, A. Ebert, and J. Meyer, editors, *Human-Centered Visualization Environments*, pages 13–75. Springer, 2007.

[17] S. Luz and M. Masoodian. Visualisation of Parallel Data Streams with Temporal Mosaics. In *Int. Conf. Information Visualization (IV '07)*, pages 197–202. IEEE, July 2007.

[18] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. LifeLines: Visualizing Personal Histories. In *Proc. of Conf. on Human Factors in Computing Systems (CHI96)*, pages 221–227. ACM, 1996.

[19] W. Prinz. NESSIE: An Awareness Environment for Cooperative Settings. In *Proc. of the Europ. Conf. on Comp. Supported Cooperative Work (ECSCW'99)*, pages 391–410. Kluwer Academic Publishers, 1999.

[20] B. Shneiderman and C. Plaisant. *Designing the User Interface*. Addison Wesley, 4th edition, 2004.

[21] J. J. Thomas and K. A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE, 2005.

[22] P. Toth and D. Vigo. *The Vehicle Routing Problem*. SIAM, 2002.

[23] M. W. van Someren, Y. F. Barnard, and J. A. Sandberg. *The Think Aloud Method: A Practical Guide to Modelling Cognitive Processes*. Academic Press Limited, 1994.