Inhaltsverzeichnis

1 Test **2**

1.1 Funktionstest des Gesamtprogramms 2

1.1.1 Szenario 2

1.1.2 Kunden und Dienstwünsche 3

1.1.3 Touren, Untertouren und Fahrten 3

1.1.4 Test der Menüpunkte Ressourcen, Ausgabe, Analyse und Einstellungen 3

1.1.5 Probleme unter Windows und JDK1.1.5 4

1.1.6 allgemeine Fehler 4

1.1.7 Ergänzungen zum Test 4

1.2 Grenzen des Systems: Test mit großen Datenmengen 5

1.3 Abgleich mit den Anforderungen 7

Kapitel 1

Test

1.1 Funktionstest des Gesamtprogramms

Eigentlich hätte vor Beginn der Testphase ein Schnitt gemacht werden sollen, so daß nach einem Implementierungsstop eine definierte „Version 0“ des Programms zur Verfügung gestanden hätte. Da dies durch krampfhaftes Festhalten an der Fertigstellung gewisser Programmteile so nicht realisiert werden konnte, gründete der Funktionstest nicht nur auf von Teil zu Teil unterschiedlichen Versionen, sondern stellte teilweise auch Fehler fest, die mittlerweile längst behoben waren. Da es sich hierbei teilweise um wichtige Programmteile handelt, sollen hier kurz diejenigen Punkte wiedergegeben werden, die der Funktionstest als fehlerhaft beschreibt, die aber im Programm zum Zeitpunkt des Endberichts korrigiert waren:

1.1.1 Szenario

Für die folgenden Tests wurden verschiedene Szenarien (u.A. mit DRK-Testdaten, mit generierten Testdaten und mit Daten aus 1.3) verwendet.

Die Menüpunkte *neu*, *öffnen* und *Szenario zum Masterszenario machen* zeigten bei mehrmaligen Aufrufen und auch bei zwischenzeitlichem Verlassen des Programms die gewünschte Wirkung. Das Speichern scheint jedoch Probleme zu machen (oder kam nicht immer der Hinweis „Szenario geändert! - Speichern?“)? Jedenfalls waren nicht immer alle Daten da, die zuvor eingegeben wurden. Allerdings konnte dieser Fehler bisher nicht reproduziert werden.

Auch beim Test der Funktionen des Untermenüs *TourSzenario* gab es Probleme:

wechseln zeigte keine Wirkung.

kopieren schlägt fehl, da die vom System automatisch aufgerufene Methode korrigiereFahrzeiten() zu einer Exception führte.

zum Master machen Funktioniert auf den ersten Blick, allerdings sieht man auf der Konsole, daß das unbekannte Kommando *tourszenario_fahrten_schreiben* aufgerufen wird.

Die übrigen Menüpunkte zeigten bei mehrmaligen Aufrufen und auch bei zwischenzeitlichem Verlassen des Programms die gewünschte Wirkung.

1.1.2 Kunden und Dienstwünsche

Abgesehen von den in Kapitel 1.3 genannten Fehlern fiel nichts mehr auf. Der Menüpunkt *Dienstwünsche erfüllt?* dürfte den Planer nur für kurze Zeit erfreuen: es werden angeblich immer alle Wünsche erfüllt!

1.1.3 Touren, Untertouren und Fahrten

Die Probleme, die sich hinter dem Menüpunkt *Tourenliste* verbergen, werden im Detail in 1.3 beschreiben. Insbesondere ließen sich keine Untertouren bilden. Damit war auch ein Test aller Menüpunkte/Dialoge, die sich auf Untertouren oder Fahrten beziehen, nicht möglich.

Der Konsistenztest entdeckte fehlende Mitarbeiter und Qualifikationen. Ebenso wurden zugewiesene, aber vom Kunden abgelehnte Mitarbeiter korrekt entdeckt. Auf mehrfach verplante Mitarbeiter und Fahrzeuge wurde jedoch nicht hingewiesen.

1.1.4 Test der Menüpunkte Ressourcen, Ausgabe, Analyse und Einstellungen

1.1.4.1 Vorgehensweise

In diesem Testabschnitt sollten die Eingabemasken für Mitarbeiter, Fahrzeuge und die systemweit bekannten Daten (=Einstellungen), sowie die Ausgabe- und Analysefunktionen auf ihre Funktionalität hin überprüft werden. Die Vorgehensweise für den Test der Eingabemasken war folgende: Eingabe neuer Daten (auch falsche Formate), Ändern und Löschen bestehender Daten. Für den Test der Ausgabe- und Analysefunktionen wurde auf schon bestehende Daten zugegriffen.

1.1.4.2 Ressourcen

Sowohl die Masken zur Neueingabe, zum Ändern und zum Löschen von Mitarbeitern, sowie die zur Manipulation der Fahrzeugdaten funktionieren im gewünschten Umfang. Bei der Neueingabe diverser Daten (z.B. Adressen, Kommunikationsverbindungen...) eines für das System neuen Mitarbeiters bzw. Fahrzeugs erscheint im Titel des Fensters nicht der Name des Mitarbeiters bzw. Fahrzeugs, sondern *null null*. Dies ist bedingt durch das Konzept des Zwei-Phasen-Commit, welches sämtliche Eingaben nach Drücken des OK Button erst prüft und danach dem System bekannt macht.

1.1.4.3 Ausgabe

Die Menüpunkte zur Ausgabe waren zum Zeitpunkt an dem der Test durchgeführt wurde, teilweise noch nicht vollständig implementiert und funktionierten dementsprechend noch nicht.

1.1.4.4 Analyse

Die Menüpunkte zur Anzeige der Mitarbeiterauslastung und zur Fahrzeugbesetzung zeigten die gewünschten Daten richtig an. Die weiteren Analysefunktionen waren zum Testzeitpunkt noch nicht vollständig implementiert und konnten somit nicht getestet werden. Die Funktionen zum Drucken funktionieren mit der Solaris-Version des JDK nicht (X11MOTIF-Exception).

1.1.4.5 Einstellungen

Die Masken zur Eingabe der einzelnen, systemweit bekannten Daten, funktionierten zur vollen Zufriedenheit. Auch falls die Daten in anderen Masken (z.B. Fahrzeugeingabe) verändert oder ergänzt werden. Die Eingaben falscher Datenformate (z.B. Eingabe von Buchstaben statt Zahlen) werden von den Konsistenztests sicher erkannt und erfolgreich abgewiesen.

1.1.4.6 Allgemeine Systemprobleme

Bei manchen Testläufen kam es vor, daß das System ohne Fehlermeldung *abstürzte* oder einfach *hängen* blieb und dadurch von außen beendet werden mußte. Da diese Fehler in unterschiedlichen Bediensituationen auftraten, liegt die Ursache wahrscheinlich in einer gewissen Instabilität der Java-Laufzeitumgebung.

1.1.5 Probleme unter Windows und JDK1.1.5

- Die Eingabe des Zeichens „@“ ist nicht möglich.
- Der Warte-Mauscursor wird erst nach einer Mausbewegung wieder in einen Aktiv-Mauscursor geändert. Spielt man nicht die ganze Zeit mit der Maus, kann man lange warten, bis eine Aktion beendet ist.
- Zeitrahmen werden immer um einen Tag nach vorne geschoben

1.1.6 allgemeine Fehler

- Bilden von Untertouren nicht möglich.
- Es ist nicht möglich, einen Dienstwunsch zu löschen, der von Touren referenziert wird. Es erscheint zwar die Frage „Dienstwunsch aus allen Touren entfernen?“, eine Antwort mit ok wird jedoch nur durch eine Exception quittiert.

1.1.7 Ergänzungen zum Test

Tourszenarien Tourszenarien können jetzt kopiert werden, zwischen verschiedenen Tourszenarien eines Szenarios kann gewechselt werden, und das jeweils aktive Tourszenario kann zum Master-Tourszenario gemacht werden (notwendige Bedingung für das Schreiben von Fahrten).

Untertouren erstellen Untertouren konnten erstellt werden und mit Teildienstwünschen bestückt werden. Das Verschieben einzelner Teildienstwünsche klappte ebenso wie das Verschieben von Halten, wobei die Prüfung auf richtige Reihenfolge der Stationen korrekt arbeitete. Da im Modus ohne Verkehrstool getestet wurde, wurden unbekannte Entfernungen bzw. Fahrzeiten angezeigt und konnten vom Benutzer eingegeben werden.

Fahrten erzeugen Zu den erzeugten Untertouren wurden Fahrten erzeugt. Dabei wurden die Zeiträume der Dienstwünsche beachtet und gegebenenfalls keine neuen Fahrten erzeugt. Die neu erzeugten Fahrten wurden dem Benutzer zur Überprüfung und möglichen Bearbeitung präsentiert. Hier konnten Mitarbeiter und Fahrzeug angepaßt werden, sowie Dienstwünsche entfernt und wieder eingefügt werden.

Das Erstellen von Dienstplänen auf der Basis dieser Fahrten gelang, ebenso wie das Archivieren von Fahrten in eine Log-Datei.

Erfüllung von Dienstwünschen prüfen Es wurden zwei Dienstwünsche überprüft, von denen einer teilweise in Touren und Untertouren enthalten war. Das Programm meldete korrekt alle Teildienstwünsche dieser Dienstwünsche, die nicht in den bestehenden Untertouren erfüllt wurden.

Löschen von Dienstwünschen, die in Touren enthalten sind Nach der Warnmeldung, daß der zu löschende Dienstwunsch noch in Touren steckt, wurde die Option gewählt, diesen automatisch auch dort entfernen zu lassen. Nach Eingabe einer Entfernung (die in der Untertour durch Wegfall des Dienstwunsches benötigt wurde) war der Dienstwunsch sowohl aus der Wunschliste des Kunden als auch aus der Tour und Untertour gelöscht, in denen er zuvor noch referenziert worden war.

Drucken von Analysen und Plänen Pläne und Analysedaten können jetzt auch ausgedruckt werden. Analyselisten werden dabei in Seiten umbrochen, von Dienstplänen wird jeweils nur der Teil ausgedruckt, der auf eine Seite paßt (dies ist etwa eine Woche).

1.2 Grenzen des Systems: Test mit großen Datenmengen

Um die Grenzen des Systems zu erkunden, wurde ein Programm geschrieben, das große Mengen zufälliger Daten anlegt, wobei verschiedene Parameter eingestellt werden können. Die Daten werden in Szenarien gespeichert, die dann von TROSS geladen werden können. Mit diesem Programm wurden zwei Szenarien erzeugt:

1. Das erste Szenario enthält je 50 Qualifikationen, Essensarten, Hilfsmittel und Fahrzeugtypen, je 200 Fahrzeuge und Mitarbeiter, 500 Kunden mit jeweils 0 bis 2 Dienstwünschen, 10 Schulen, 150 Schultouren mit 3 bis 6 Kindern und die Untertouren zu den Schultouren (eine Hin- und eine Rückfahrt).

2. Im zweiten Szenario wurde die Zahl der Kunden auf 1000 erhöht, die Zahl der Schulen auf 20 und die Zahl der Schultouren auf 300.

Diese Szenarien wurden geladen und verschiedene Funktionen von TROSS aufgerufen, um die Schwachstellen zu finden. Für diese wurden dann Zeitmessungen durchgeführt. In Tabelle 1.1 werden die nacheinander aufgerufenen Funktionen und die benötigten (gestoppten) Zeiten in Sekunden auf zwei Rechnern (beide Windows 95 mit JDK 1.1.6) angegeben.

Funktion	Cyrix 166/32MB	P100/24MB
Szenario 1 laden	115	319
Kunden anzeigen	5	12
speichern als Szenario 3	100	152
Szenario 3 laden	110	307
Kunden anzeigen	2,5	8
Szenario 2 laden	305	912
Kunden anzeigen	8	18
Touren anzeigen	2	5

Tabelle 1.1: Zeiten für verschiedene Funktionen

Das große Problem: das Laden (und Speichern) der Daten. In TROSS wird die Serialisation von Java verwendet, wobei die Daten noch komprimiert werden. Auch das Abschalten der Komprimierung änderte kaum etwas. Der Profiler ermittelt für das Laden des kleinen Szenarios die folgenden Top-Rechenzeit-Verbraucher:

59,4%	ObjectInputStream.inputClassFields(Object, Class, int[])
20,9%	System.gc()
11,2%	Object.wait(long)

Alle drei Methoden kommen von Java. Die erste Methode ist nicht dokumentiert, dem Namen nach liest sie die Attribute der Objekte. Die zweite ist der Garbage-Collector und die dritte dient zur Synchronisation paralleler Zugriffe. Eine Laufzeit-Optimierung ist hier also nicht möglich, am einfachsten können die 21% für den Garbage-Collector durch mehr Speicher verringert werden.

Die übrigen gestoppten Zeiten wären auf dem Cyrix-Prozessor gerade noch hinnehmbar, das Laden und Speichern mittels Objekt-Serialisation ist aber nicht akzeptabel, kann aber wegen Zeitmangels nicht mehr geändert werden.

Für die folgenden Versuche wurde ein viertes Szenario mit vom zukünftigen Benutzer von TROSS gelieferten Daten erstellt. Dieses enthält 46 Fahrzeuge, 63 Mitarbeiter, 94 Kunden mit Dienstwünschen (Essen auf Rädern und Schulfahrten) und 11 Schultouren.

Bei ausgeschaltenem Verkehrstool wurden die Dienstwünsche angelegt. Danach wurde das Verkehrstool eingeschaltet. Für das Anbinden der 95 Stationen (94 und eine Schule) wurden 95 Sekunden benötigt. Danach wurden für alle 11 Schultouren die Entfernungen zwischen den einzelnen Stationen ermittelt. Das dauerte 160 Sekunden. Hier wurden etwa $160/11 = 15$ Sekunden pro Map&Guide Auftrag.

Der benötigte Speicherplatz auf der Festplatte hält sich dank der Kompression in Grenzen. Tabelle 1.2 zeigt die Dateigrößen in Bytes verschiedener Szenarien. Weshalb das Szenario 3 kleiner ist als Szenario 1, ist nicht erklärbar.

Die Szenarien in den letzten drei Zeilen enthalten jeweils noch die Fahrten der Schultouren für 1, 2 bzw. 3 Wochen. Pro Fahrt werden etwa $(17905 - 17187)/220 = 3,2$ Bytes benötigt. Dank der Kompression gibt es hier also keine Engpässe.

Szenario	Größe in Bytes
Szenario 1	134.472
Szenario 2	254.954
Szenario 3	134.423
Szenario 4	16.292
Szenario 4 + 1 Woche	17.187
Szenario 4 + 2 Wochen	17.513
Szenario 4 + 3 Wochen	17.905

Tabelle 1.2: Größen verschiedener Szenarien

1.3 Abgleich mit den Anforderungen: Durchspielen der Szenarien

Ein Teil der Testphase bestand darin, die Daten aus den Szenarien der Anforderungsanalyse mit TROSS zu verarbeiten. Im folgenden soll dargestellt werden, welche Teile mit TROSS verarbeitet werden konnten, welche sich aufgrund von Fehlern nicht verarbeiten ließen und welche aus konzeptionellen Gründen nicht mit TROSS verarbeitet werden konnten.

1.3.0.1 Schulfahrten

Szenario 1 Die Qualifikationen „Lastenheben“ und „Epilepsie-Umgang“ wurden neu angelegt. Daraufhin wurden die Zivildienstleistenden Gustav Häberle und Karl Tremmle gemäß den Vorgaben des Szenarios erfaßt. Danach wurden die vier vorgegebenen Kunden erfaßt und zu einer Tour zusammengestellt, die wie gefordert für die automatische Optimierung gesperrt wurde. Da die Schülerin Karin Klein krank ist, wurde in ihrem Dienstwunsch die voraussichtliche Dauer der Krankheit vom Zeitrahmen ausgenommen.

Probleme

- Da für Touren keine Bemerkung vorgesehen ist, muß der Auftrag, die Schüler in die Klassen zu bringen, bei jedem Dienstwunsch angegeben werden.
- Der Zivi Karl Tremmle machte zu Beginn seiner Dienstzeit einen Lehrgang. Dies sollte in TROSS durch eine Datumsspanne für die Zeit des Lehrgangs dargestellt werden. Dazu wäre es wünschenswert gewesen, bei den Ausnahemzeiträumen auch Bemerkungen eingeben zu können, damit man weiß, warum ein Mitarbeiter fehlt. Bisher kann nur dargestellt werden, daß ein Mitarbeiter nicht verfügbar ist, aber nicht, warum er fehlt.
- Daß der Klassenlehrer von Karin Klein über deren Krankheit unterrichtet werden soll, ist im System nicht sinnvoll darstellbar, da der entsprechende Dienstwunsch nicht mehr in der Fahrt auftaucht.
- Die Zeitangabe ohne Minuten („16: “) führte zu der verwirrenden Fehlermeldung „ganze Zahl eingeben“.

Fehler

- Beim Erstellen von Untertouren trat eine Exception auf. Daher konnte auch nicht festgestellt werden, ob es möglich ist, daß die Schülerin Susanne Schlecht auf der Hinfahrt garantiert als letzte abgeholt und bei der Rückfahrt als erste abgeliefert wird.

Szenario 2 Das Szenario konnte mit dem System komplett umgesetzt werden. Ob die Änderungen der Dienstwünsche in den Touren durch Umsetzen der Kunden in den Untertouren und Fahrten richtig nachvollzogen wird, konnte nicht festgestellt werden, da es nicht möglich war, Untertouren zu bilden.

1.3.0.2 MSD / Pflegedienst

Szenario 1 Das Szenario konnte mit TROSS nachvollzogen werden. Die Vorgabe, beide Dienstanforderungen als einen Dienstwunsch zu erfassen, kann nachvollzogen werden, hat im System jedoch folgendes Problem: Auf der Tour muß immer ein teurer Pfleger mitgehen, auch wenn nur Einkäufe zu erledigen sind.

Szenario 2 Bei der Umsetzung des zweiten MSD-Szenarios traten zwei Fehler auf. Zum einen war gefordert, einen bevorzugten Mitarbeiter festzulegen. Dies geschah bei der Eingabe des Dienstwunsches, allerdings war die Eingabe beim nächsten Aufruf der Dienstwunschmaske wieder verschwunden.

Das zweite Problem machte die Ausnahmezeiterfassung beim Kunden. Es sollten zwei Zeiträume eingegeben werden, in denen die Kundin nicht bedient werden soll. Allerdings wurde die zweite Eingabe ignoriert und dafür der erste Zeitraum (unter Windows) um einen Tag nach vorne verschoben.

1.3.0.3 Essen auf Rädern

Szenario 1 Das Szenario konnte erfolgreich nachvollzogen werden.

Szenario 2 Bei diesem Szenario traten folgende Probleme auf:

Zunächst sollte eine Email-Adresse eingegeben werden, da der Kunde per Email bestellte. Dabei fiel auf, daß unter Windows95 die Eingabe des Zeichens „@“ nicht möglich ist, obwohl dies unter UNIX funktioniert.

Das nächste Problem war die nachträgliche Adreßerfassung in der Dienstwunschmaske. Da zunächst keine Lieferadresse angegeben war, erschien korrekterweise eine entsprechende Fehlermeldung. Daraufhin wurde die Lieferadresse ergänzt. Diese konnte vom Verkehrsmodul nicht angebunden werden, weshalb ein entsprechender Hinweis kam, der mit Abbrechen beendet werden sollte. Leider blieb daraufhin auch TROSS stehen und mußte von außen verlassen werden.

Nachdem der Dienstwunsch wegen obigem Fehler nochmals erfaßt war, sollte sichergestellt werden, daß er am Ende einer Essenstour erledigt wird. Eine Reihenfolge kann jedoch erst bei Untertouren festgelegt werden, die wie schon erwähnt nicht funktionierte. Damit bleibt dieser Punkt ungeprüft. Als Alternative wurde eine späte Lieferzeit eingegeben, die jedoch nichts garantiert.

Die Verquickung mit einem MSD-Dienstwunsch ist aus konzeptionellen Gründen nicht direkt möglich. Dies muß bei der Tourplanung manuell berücksichtigt werden (wobei die beiden Dienstwünsche wegen der unterschiedlichen Dienstarten nicht in einer Tour erledigt werden dürfen!). Eine andere Möglichkeit wäre ein MSD-Dienstwunsch mit der Bemerkung „Essen mitnehmen“ oder ein EAR-Dienstwunsch mit einer längeren Aufenthaltszeit. Dies bringt aber Probleme für die (in dieser Version nicht realisierte) automatische Abrechnung mit sich, da sich diese auf die Dienstart stützt und somit entweder ein Essen oder MSD-Stunden in Rechnung stellen könnte.

Szenario 3 Beim Szenario 3 ergeben sich durch die geforderte Verbindung von EAR- und MSD-Dienst die schon oben genannten Probleme. Weitere Probleme traten in Verbindung mit der Erfassung von Institutionen auf:

- Obwohl die Klasse Kunde einen Vektor für die Aufnahme von Institutionen besitzt, ist es ist nicht möglich, dem Kunden direkt, d.h. ohne bekannten Sachbearbeiter, eine Institution zuzuweisen. Ist der Sachbearbeiter unbekannt, muß die Institution über eine Dummy-Bezugsperson referenziert werden.
- Es ist keine Angabe der Versicherten-Nummer vorgesehen. Diese kann zwar im Feld Bemerkung untergebracht werden, verhindert aber (z.B. für automatische Rechnungsstellung) eine Auswertung durch das Systems.

Szenario 4 Das Szenario konnte mit TROSS erfaßt werden. Ein Problem trat unter Windows95, jedoch nicht bei UNIX auf: Der Zeitrahmen wird beim Verlassen der Zeitrahmeneingabe um einen Tag nach vorne verschoben.

Szenario 5 Das Szenario 5 konnte problemlos nachvollzogen werden.

Szenario 6 Der entsprechende Ausnahmezeitraum wurde erfaßt (und nach Szenario 4 erwartungsgemäß verschoben). Auswirkungen auf die Fahrten konnten nicht getestet werden, da es nicht möglich war, Fahrten zu erzeugen.

Szenario 7 Der nicht mehr benötigte Dienstwunsch wurde durch Angabe eines Enddatums in seiner Zeitspanne suspendiert. Die Entfernung aus der entsprechenden Tour muß manuell erfolgen, Auswirkungen auf die Untertouren und Fahrten konnten nicht getestet werden.

Szenario 8 Solche kurzfristigen Änderungen werden vermutlich nicht über das System laufen. Falls doch, müssen die entsprechenden Fahrten geändert werden (2 Fahrten zu einer Untertour, wobei jede einen Teil erledigt). Dies konnte nicht getestet werden, da es nicht möglich war, Fahrten zu erzeugen.

1.3.0.4 Individualfahrten

Szenario 1 Dieses Szenario (Kundin fährt zum Arzt und wieder zurück) kann problemlos nachvollzogen werden.

Szenario 2 Bei der Eingabe dieses Szenarios einer Individualfahrt mit einer Zwischenstation blieb die Frage offen, wann die Zwischenstation im System berücksichtigt wird (auf der Hinfahrt, auf der Rückfahrt, immer?). Außerdem kann nur angeben werden, ob der Mitarbeiter zwischen der Hin- und Rückfahrt anwesend sein muß, für Zwischenstationen ist keine Angabe möglich. Eine andere Möglichkeit dieses Szenario so einzugeben, daß der Mitarbeiter dem DRK zwischen allen Stationen zur Verfügung steht, wäre die Aufspaltung des Dienstwunsches auf drei Individualfahrten (Wohnung, Friedhof), (Friedhof, Gemeindehaus), (Gemeindehaus, Wohnung).

Außerdem war es wegen Exceptions nicht möglich, überhaupt Zwischenstationen anzugeben.

Szenario 3 Entspricht, abgesehen von der Anfangszeit, dem Szenario 2.

Szenario 4 Ist für die Dateneingabe völlig uninteressant, da die Fahrt abgelehnt wird. Diese Entscheidung kann auf Grundlage der Datenausgabe getroffen werden, deren Testergebnisse in 1.1.4.3 zu finden sind.