

TINF

Editor und Simulator
für
Datenflußdiagramme

Bedienungshandbuch

Fortgeschrittenen-Praktikum
von
Jochen Langbein
und
Christian Schurr

langbein@informatik.tu-muenchen.de
schurr@informatik.tu-muenchen.de

im Sommersemester 1996
an der Technischen Universität München

am Lehrstuhl von
Prof. Dr. Manfred Broy

Betreuung durch
Bernhard Rumpe

Inhaltsverzeichnis

1. EINLEITUNG	3
2. DATENFLUSSDIAGRAMME, BEGRIFFE	4
3. BEDIENUNGSKONZEPT	6
3.1 DER EDITOR	6
3.2 DER SIMULATOR	8
4. TUTORIAL	9
5. DAS ARBEITEN MIT TINF	22
5.1 EDITOR	23
5.2 SIMULATOR	30
6. VORDEFINIIERTE OBJEKTE	34

1. Einleitung

TINF ist ein einfach zu bedienender, interaktiver Datenflußeditor und -simulator. Die Kombination der beiden Funktionalitäten in einem Programm ermöglicht die schnelle Erstellung von vereinfachten Datenflußdiagrammen und deren Simulation am Bildschirm. Das Erzeugen der Datenflußdiagramme erfolgt interaktiv am Bildschirm mit Hilfe der Maus. Die platzierten Objekte entsprechen beliebigen GOS-Objekten, die der Benutzer vorher definiert hat bzw. schon in der Standardbibliothek vorhanden sind. Diese Objekte haben eine festgelegte Schnittstelle und sind einfach vom Benutzer zu erzeugen und zu gebrauchen. Die Objekte werden über Leitungen miteinander verbunden. Die Simulation des aktuellen Datenflußdiagramms kann zu jeder Zeit erfolgen, da das Programm nur korrekte Verbindungen zuläßt.

Das Kapitel „Bedienungskonzept“ erläutert die generelle Arbeitsweise mit TINF. Es ist für ein effektives Arbeiten daher ratsam, diesen Abschnitt zu lesen.

Das „Tutorial“ bietet Einsteigern eine geleitete Führung durch die wesentlichen Funktionen. Einzelne Arbeitsschritte können schließlich bei Bedarf im Kapitel „Das Arbeiten mit TINF“ nachgelesen werden.

2. Datenflußdiagramme, Begriffe

In diesem Kapitel wird geklärt, was in diesem Bedienungshandbuch unter Datenflußdiagrammen im Zusammenhang mit diesem Editor und Simulator verstanden wird.

Allgemeine Grundlagen über Datenflußdiagramme stehen hier nicht zur Diskussion und Kenntnisse darüber werden als bekannt vorausgesetzt.

Datenflußdiagramme werden zu Präsentations- und Erläuterungszwecken oft auf dem Papier oder in herkömmlichen, vektororientierten Graphikprogrammen gezeichnet.

Der Designer unterliegt deshalb keinen Einschränkungen bezüglich seiner gestalterischen Fähigkeiten und das Resultat kann ein stimmiges Diagramm sein oder aber „syntaktische“ Fehler enthalten. (Damit sind Konstellationen gemeint, die in einem eigentlichen Datenflußdiagramm nicht auftreten sollten.) Dies bedeutet nicht, daß es nicht von Fall zu Fall durchaus seine Berechtigung hat, und zum Beispiel bestimmte Sonderfälle veranschaulicht.

TINF benutzt das gezeichnete Diagramm hingegen um daraus eine lauffähige Simulation zu erstellen. So ergibt es sich automatisch, daß einige Zusatzbedingungen bezüglich des Aussehens gebraucht werden, um Diskrepanzen zwischen dem, was der Designer will, und dem, wie es TINF interpretiert, von vornherein zu vermeiden.

Einige andere Einschränkungen dienen lediglich dazu, die Bedienung einfach und das Programm nicht übermäßig kompliziert werden zu lassen.

In Datenflußdiagrammen treten als aktive Komponenten *Prozesse* und *Speicher* auf. Speicher sind (bezüglich einer Simulation) in ihrer Funktion kaum präzisiert und stellen quasi einen Prozeß dar, der Daten speichert und wieder zur Verfügung stellt. Deshalb werden sie von TINF wie Prozesse behandelt und unterscheiden sich lediglich dem Aussehen nach.

Prozesse sind zur einfacheren Handhabung Rechtecke, jedoch mit abgerundeten Ecken um dem kreisförmigen Pendant der Standard-Datenflußdiagramme zu ähneln. Speicher werden wie gewohnt als zwei waagrechte Striche gezeichnet, und *Ein-* und *Ausgänge* eines Diagramms durch Rechtecke repräsentiert.

Da nun alle vier genannten Symbole rechteckige Flächen abdecken und graphisch auf die gleiche Weise bearbeitet werden, wird im Zusammenhang mit dem Editor nur noch von *Boxen* die Rede sein.

Die Verbindungen zwischen Boxen werden als *Leitungen* bezeichnet. Sie setzen sich aus einzelnen *Kabeln* oder *Leitungsstücken* zusammen, die über *Knoten*, gekennzeichnet als dickere Punkte, verbunden sind. Die Enden der Leitung werden *Start-* bzw. *Endknoten* genannt.

Nun lassen sich die möglichen TINF-Diagramme also als Untermenge der Datenflußdiagramme beschreiben.

Ein Datenflußdiagramm ist „korrekt“ im Sinne von TINF, wenn folgende Aussagen gelten:

- Leitungen haben genau einen Anfang (gekennzeichnet durch einen dickeren Knoten) und beliebig viele Enden.
- Leitungen enthalten keine Zyklen.
- Leitungen beginnen und enden nur am Rand von Speichern und Prozessen oder an einer nicht belegten Stelle.
- Leitungsstücke dürfen keine Knoten überschneiden.
- Symbole von Prozessen und Speichern dürfen sich nicht gegenseitig überlappen und nicht von Leitungen geschnitten werden.

3. Bedienungskonzept

3.1 Der Editor

Die mit dem Editor erstellten Diagramme sollten zu jedem Zeitpunkt des Erstellens in einem „korrekten“ Zustand sein, bzw. sich zu einem solchen vervollständigen lassen können.

Dies schließt also auch während der Konstruktion alle Konstellationen von Leitungen und Boxen aus, die in einem gültigen, fertiggestellten Diagramm nicht vorkommen dürfen. Diese wurden im vorhergehenden Kapitel bereits erwähnt.

Beim Erstellen von Diagrammen wird die Arbeit also im allgemeinen an drei Arten von graphischen Objekten verrichtet: An Boxen, Leitungen und Beschreibungstexten.

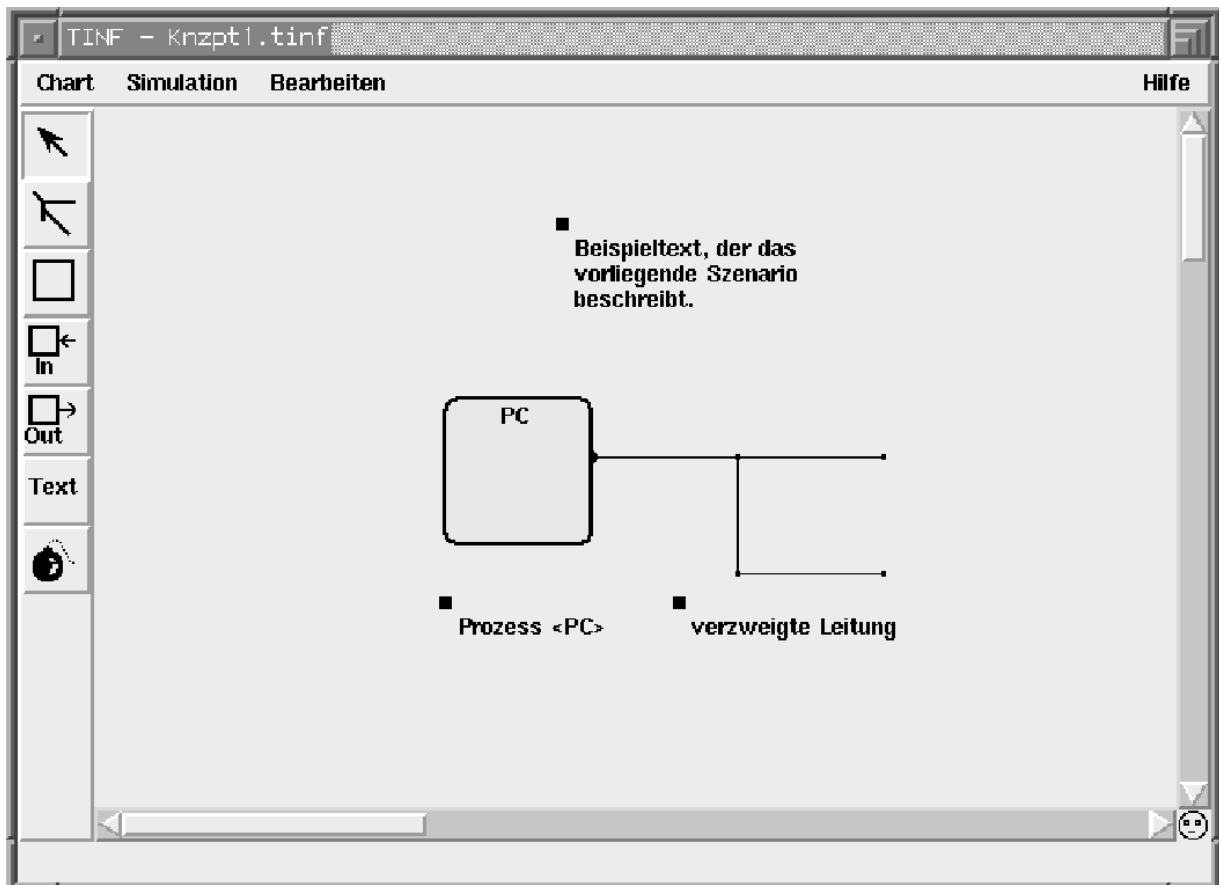


Abb. 3.1: TINF mit Beispielobjekten

Um neue Objekte auf den Bildschirm zu setzen, wird der entsprechenden Modus gewählt und die Objekte an der gewünschten Stelle plziert. Einfache nachträgliche Änderungen, etwa der Position, sind unabhängig vom eingestellten Modus durchführbar. Für spezielle Änderungen, die mit dem Kreieren neuer Objekte nichts zu tun haben oder zum großflächigen Auswählen

von Objekten, dient der Markier/Editiermodus, gekennzeichnet durch den Pfeil. Die genauen Funktionen werden im Kapitel 5 erläutert.

Beim Arbeiten mit Prozessen und Speichern sollte sich der Benutzer im klaren sein, daß diese echte GOS-Objekte repräsentieren, deren sprachliche Beschreibung der Klasse, von der sie instanziiert werden, bestimmte Dinge voraussetzen. So zum Beispiel die Anzahl der Ein- und Ausgänge. TINF verhindert nur, daß zu viel Ein- oder Ausgänge angeschlossen werden, da die Unterbelegung für einige Objekte sinnvoll sein kann.

Boxen können verschoben, in ihrer Größe verändert und (wie alle Objekte) natürlich gelöscht werden.

Die Boxen werden, wie bereits erwähnt, durch Leitungen verbunden.

Die Leitungen erhalten (später bei der Simulation) ihre Information genau aus einer Box und können diese an beliebig viele weitergeben. Daher sind sie als Baumstruktur organisiert, mit dem „Eingang“ (Startknoten) als Wurzel, was Zyklen von vornherein ausschließt.

Auch folgt daraus, daß Leitungen nicht direkt verbindbar sind, ausgenommen einer der Verbindungspunkte ist ein Startknoten, wodurch natürlich wieder ein einziger Baum mit nur einer Wurzel entstehen würde.

Die Bearbeitungsmöglichkeiten der Leitungen sind sehr umfangreich. Sie beziehen sich entweder auf die ganze Leitung, Teilbäume der Leitung, einzelne Knoten oder auf Kabel.

Die gesamte Leitung sowie Teilbäume davon können gelöscht und verschoben werden, wobei der Teilbaum möglicherweise abgetrennt und zu einer eigenständigen Leitung wird.

Knoten lassen sich nur löschen, wenn sich dadurch die Struktur der Leitung nicht verändert, wie etwa bei einem Zwischenknoten innerhalb eines geraden Stückes, oder aber wenn sie sich am Anfang oder Ende des Baumes befinden. Nur Endknoten können einzeln verschoben werden.

Löschen eines Kabels trennt die Leitung in zwei separate Leitungen auf.

Texte dürfen als einzige alle anderen Objekte und sich gegenseitig überlappen. Lediglich der Anfasspunkt muß an einer freien Stelle zu liegen kommen.

Alle Editier-Aktionen des Benutzers beziehen sich auf das gerade mit der Maus angewählte Objekt, das dabei optisch hervorgehoben wird.

Zur Beschleunigung des Diagrammerstellens können einige Funktionen natürlich auch an mehreren Objekte gleichzeitig ausgeführt werden. Dazu zählen zum Beispiel Verschieben und Löschen. Um mehrere Objekte zu markieren und bei Bedarf wieder zu demarkieren hält man die ‘*Shift*’-Taste beim Klicken mit der Maus gedrückt. Diese Möglichkeit existiert nicht für einzelne Knoten oder Kabelstücke.

Das Halten der ‘*Shift*’-Taste kann auch mit dem Anwählen von Teilbäumen, ausgeführt durch einen Doppelklick auf den obersten Knoten des Teilbaumes, kombiniert werden.

3.2 Der Simulator

Der Simulator beschränkt sich im wesentlichen darauf, die Informationen gemäß den Leitungen zwischen den Boxen weiterzureichen und die zugehörigen GOS-Objekte damit (als Parameter) aufzurufen. Die Ergebnisse werden schließlich wieder auf die Leitungen gelegt.

Ein Simulationsschritt holt also die Daten von den Leitungen, ruft damit reihum alle Prozesse und Speicher auf und legt die Ergebnisse (hinterher) wieder auf die Leitungen.

Da jede Art von zu simulierenden Prozessen und Speichern, abgebildet auf eine Klasse, gleich behandelt wird, werden sie alle von der gleichen Basisklasse SIMBAR abgeleitet. Sie bieten somit eine einheitliche Schnittstelle, die sich leicht aus dem vorher beschriebenen Simulationsschritt ergibt.

Abgesehen vom Konstruktor NEW gibt es je eine Methode für das Setzen der Eingänge, für den Simulationsschritt, das Auslesen der Ausgänge und zusätzlich für das Auslesen von Informationen, die der Prozeß oder Speicher evtl. anzeigen möchte.

Als Parameter und Rückgabewerte werden Listen von Daten verwendet. Die Daten sind vom Typ „String“.

Um einen Simulationsschritt auszuführen, ist der Menüpunkt „Schritt“ im Simulationsmenü auszuwählen, wobei beim ersten Aufruf noch keine Simulation erfolgt, sondern zunächst alle Prozesse und Speicher initialisiert werden.

Der Simulationsvorgang ist dann durch das Setzen und Ändern von Daten an den Eingängen und - etwas unsauber - zu Testzwecken auch an den Leitungen, beeinflussbar. Der hierzu benötigte Dialog wird über den jeweils entsprechenden Menüpunkt im Kontextmenü der Eingänge und Leitungen aktiviert. Das Kontextmenü erscheint durch Betätigen der rechten Maustaste auf einem Objekt.

4. Tutorial

In diesem Tutorial werden die grundlegenden Arbeitsschritte an Hand von Beispielen erläutert. Der Leser wird Schritt für Schritt von Bildern begleitet, die ihm die Vorgänge verdeutlichen. Das Tutorial ist als Führung mit gleichzeitiger Benutzung des Programms TINF gedacht, kann aber wegen der zahlreichen Bilder auch für sich gelesen werden.

1. Schritt: Programmstart

Das Programm wird an der Shell gestartet. Wechseln Sie nun in das Verzeichnis, in das Sie die TINF-Dateien 'tinf' und evtl. 'SIMBAR.gos' (aus dem öffentlichen TINF-Verzeichnis unter 'pub/bin/' bzw. 'pub/lib/', vgl. pub/README) kopiert haben und rufen sie das Shell - skript „tinf“ auf.

Dadurch wird die Datei SIMBAR.gos, die die Programmcodes zu den vordefinierten und den vom Benutzer hinzugefügten Prozessen und Speichern, sowie deren Verwaltungsklasse enthält, zum restlichen TINF-Programmcode kopiert und der GOS-Interpreter gestartet.

Machen Sie sich mit den Elementen im Fenster vertraut.

Im Titelfeld des Fensters erscheint neben dem Programmnamen auch der Name der Datei, die gerade bearbeitet wird (beim Programmstart also noch keiner). Das Aussehen des Titelfelds hängt natürlich vom jeweiligen X-Window-Manager und dessen Einstellungen ab.

Gleich darunter befindet sich das Menü, mit den Datei- bzw. Bearbeitungsfunktionen und rechts das Hilfenmenü.

Links befinden sich in einer Button-Leiste die einzelnen Bearbeitungsmodi und der Löschknopf (Bombe).

Beachten Sie, daß am unteren Fensterrand jeweils eine kurze Erklärung zu dem Button unter der Maus bzw. später auch zum aktuellen Bearbeitungsvorgang gegeben wird.

2. Schritt: Erste Objekte plazieren

Klicken Sie nun mit der linken Maustaste auf das Viereck an der linken Button-Leiste. Dadurch wählen Sie den Modus zum Erzeugen neuer Boxen aus.

Drücken Sie nun die Maustaste (ohne weitere Angaben sei immer die linke gemeint) auf der Arbeitsoberfläche und halten Sie sie gedrückt.

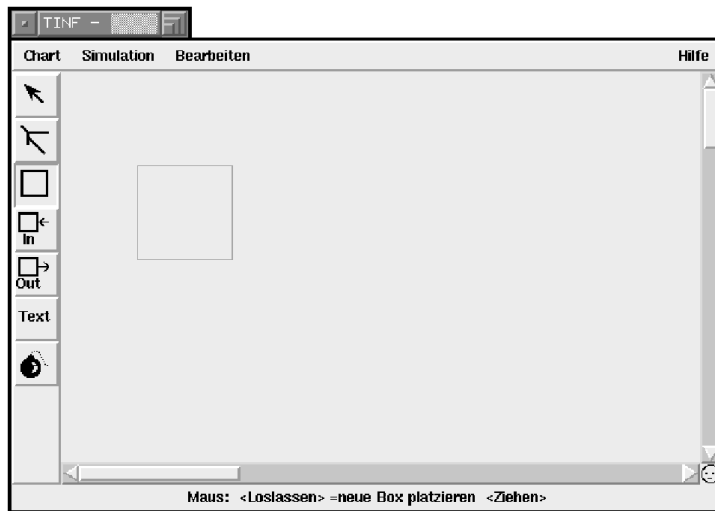


Abb. 4.1: Erstellen einer neuen Box

Es erscheint ein Viereck, das bewegt und durch Loslassen der Maustaste platziert werden kann.

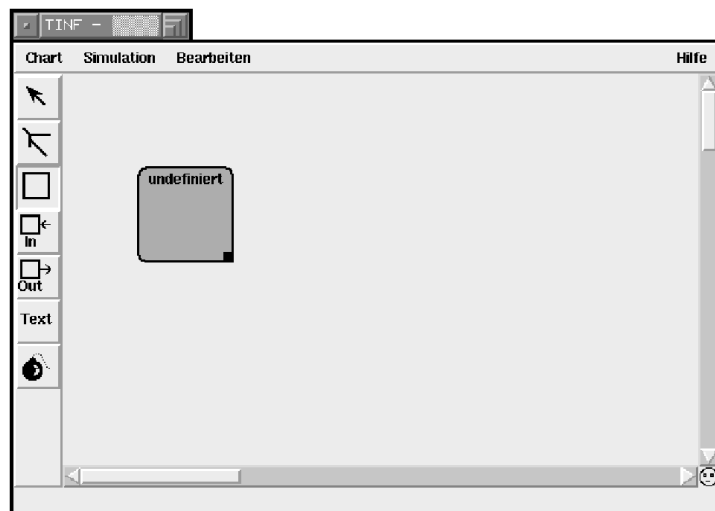


Abb. 4.2: Neue, undefinierte Box wurde erstellt

Die neu erstellte Box ist noch undefiniert, ihr wurde also keine Beschreibungsdatei (*.tinf) zugewiesen, die Auskunft über Namen, minimale Größe und Typ gibt.

Dies geschieht durch Drücken der rechten Maustaste auf der Box und anschließendem Auswählen des Menüeintrags „Typ ändern“ mit der linken Maustaste.

(Mit „Info“ können interne Informationen über die Box abgefragt und mit dem zweiten Menü - eintrag das Menü abgebrochen werden.)

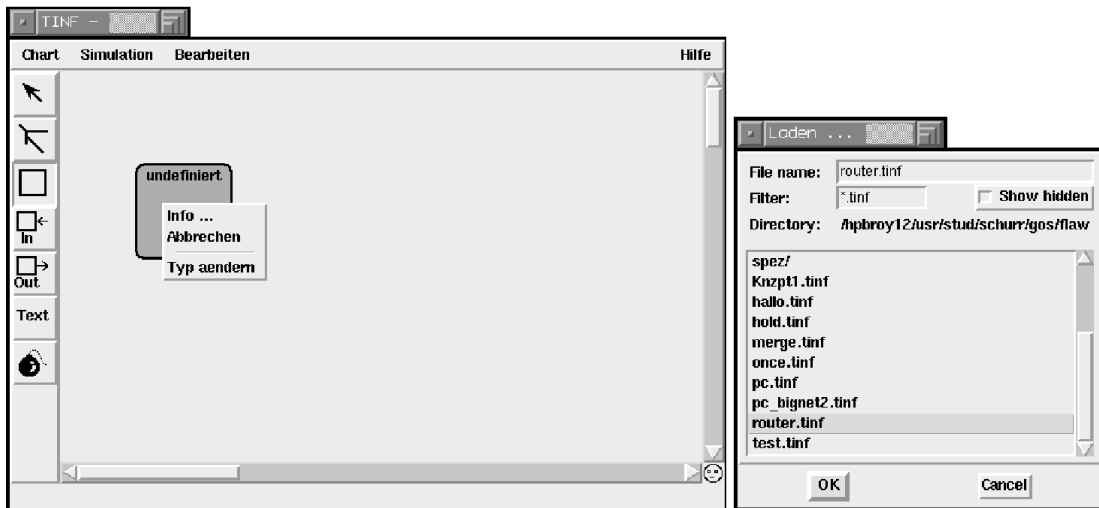


Abb. 4.3: Zuweisen einer Beschreibungsdatei

Daraufhin erscheint eine Dialogbox zum Laden von Dateien, in der Sie eine Beschreibungsdatei auswählen können. In diesem Tutorial soll dies „router.tinf“ sein. An dieser Stelle können auch ursprünglich als Diagramm gespeicherte Dateien sozusagen als Blackbox eingebunden werden. Die Anzahl der Ein-/ Ausgänge des Diagramms entspricht dann der Anzahl der Ein-/Ausgänge der Box. Das Simulieren solcher Diagramme bleibt allerdings noch späteren Versionen von TINF vorbehalten.

Als nächstes soll eine aus der Box herausführende Leitung erstellt werden. Schalten Sie dazu auf den Leitungsmodus um. Wenn Sie nun die Maustaste auf dem Rand der Box gedrückt halten und die Maus ein Stück nach rechts bewegen, sollten sich etwa folgendes Bild ergeben:

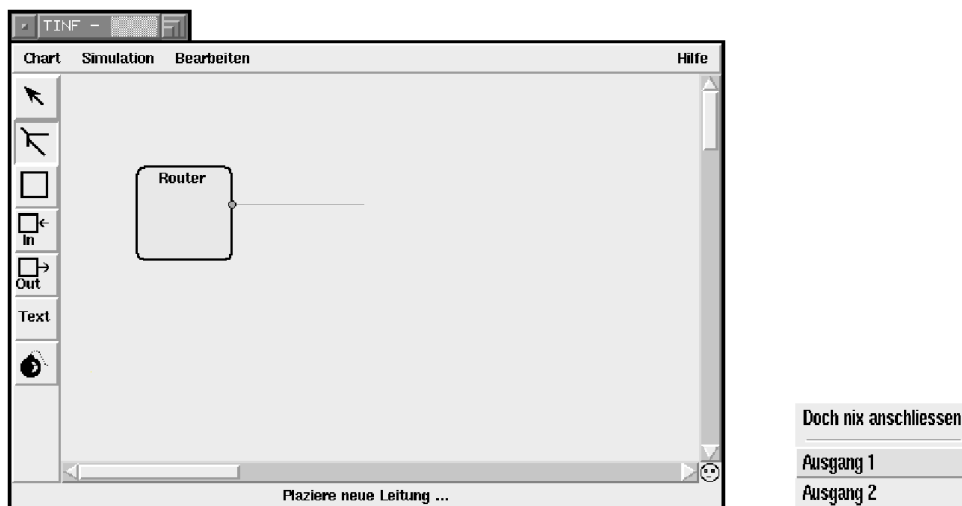


Abb. 4.4: Neue Leitung am Ausgang einer Box erzeugen

Durch Loslassen der Taste wird die Leitung verlegt, sofern sie wie hier keine anderen Boxen oder Leitungsknoten überlagert. Weil die Leitung von einer Box ausgehend gezeichnet wurde, meldet sich diese und fragt über ein Auswahlmü an welchem der Ausgänge der Anschluß

erfolgen soll. Dieses Menü erscheint nicht, wenn die Box über unendlich viele Anschlußmöglichkeiten (gekennzeichnet durch eine -1 in der Beschreibungsdatei) oder über genau einen verfügt.

Sie können die Leitung nun durch analoges Vorgehen beliebig verlängern und aufzweigen. Drücken Sie dazu die Maustaste auf einem beliebigen Knoten und ziehen Sie ein weiteres Leitungsstück.

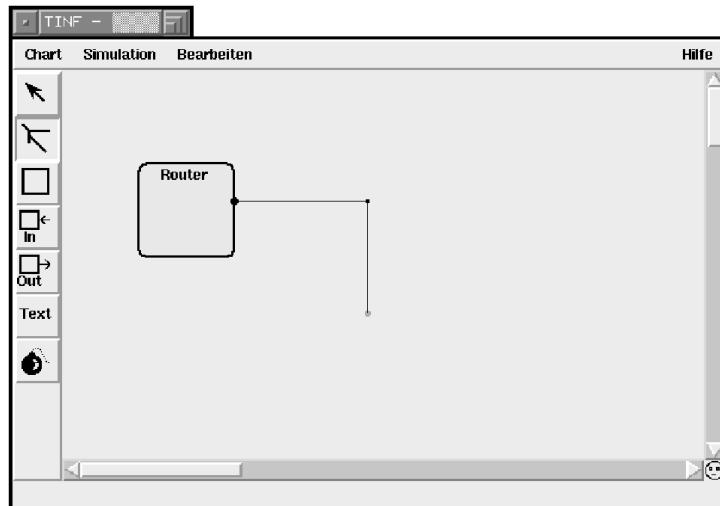


Abb. 4.5: Anfügen von Kabelstücken an die bestehende Leitung

Auf die gleiche Weise können neue Kabelstücke auch in der Mitte von anderen Kabelstücken angefügt werden. Beim Drücken der Maustaste auf einem solchen Kabelstück wird dieses zunächst einfach markiert. Beim ersten Bewegen der Maus über eine Rastereinheit hinaus wird aber bereits ein neuer Knoten eingefügt. Dieser bleibt auch bestehen, wenn kein Kabel verlegt, sondern die Maustaste genau auf diesem Knoten ausgelassen wird.

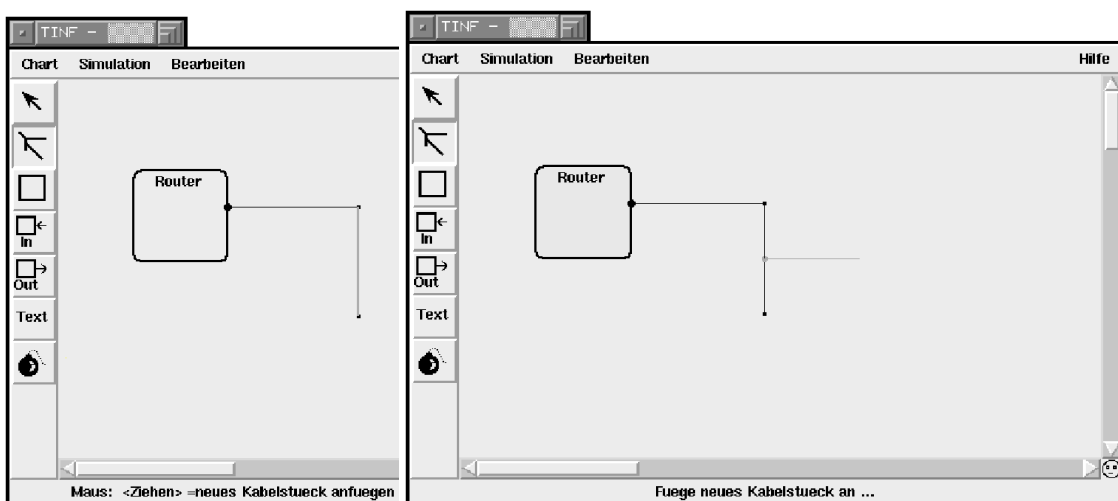


Abb. 4.6: Automatisches Einfügen eines Knotens beim Ziehen eines neuen Kabelstückes ausgehend von der Mitte eines anderen

Verlegen Sie also nun das Kabelstück wie in Abb. 4.6 dargestellt.

3. Schritt: Teilbäume manipulieren

Weitere Bearbeitungsmöglichkeit an Leitungen bietet die Manipulation von Teilbäumen. Dazu müssen diese wie gewohnt vorher markiert werden. Sie markieren einen Teilbaum durch einen Doppelklick auf dessen obersten Knoten.

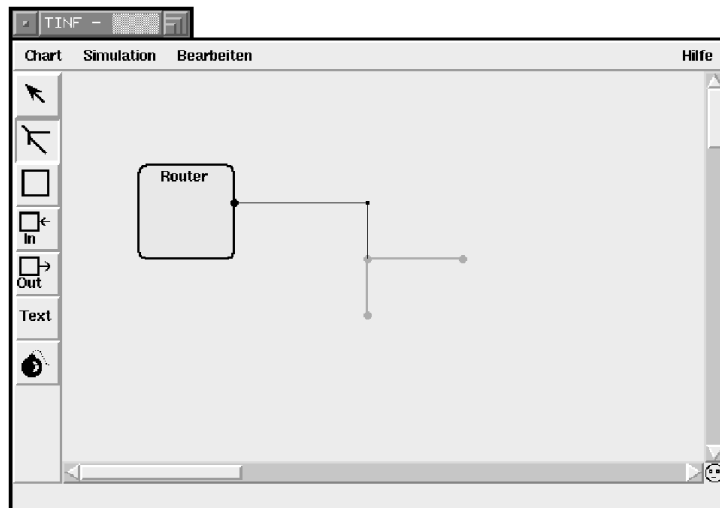


Abb. 4.7: Markieren eines Teilbaums der Leitung

Teilbäume können wie andere Objekte verschoben werden, wobei allerdings einige Dinge zusätzlich eine Rolle spielen. Je nachdem, wo ein verschobener Teilbaum zu liegen kommt, bleibt dieser mit der Leitung verbunden oder wird abgetrennt und damit zu einer eigenständigen Leitung. Dies lässt sich bereits an der schematischen Darstellung des Baumes beim Verschieben erkennen. Überall, wo waagrechte, senkrechte oder 45 Grad schräge Verbindungen möglich sind, werden diese eingezeichnet, an anderen Stellen nicht.

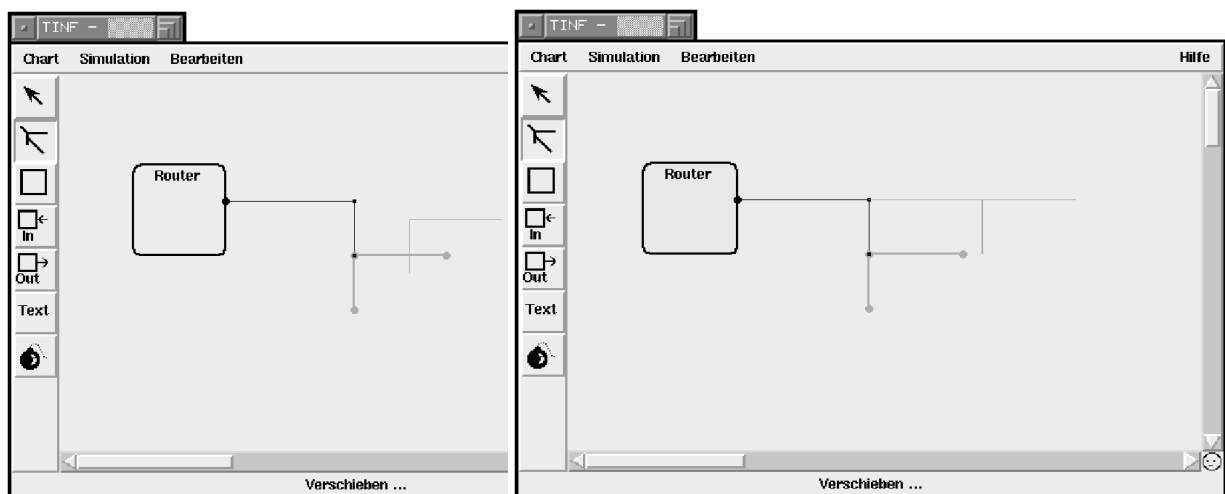


Abb. 4.8: Verschieben des Teilbaums einmal mit, einmal ohne Verbindung

Der Teilbaum soll nun so rechts von der restlichen Leitung zu liegen kommen, daß er verbunden bleiben kann.

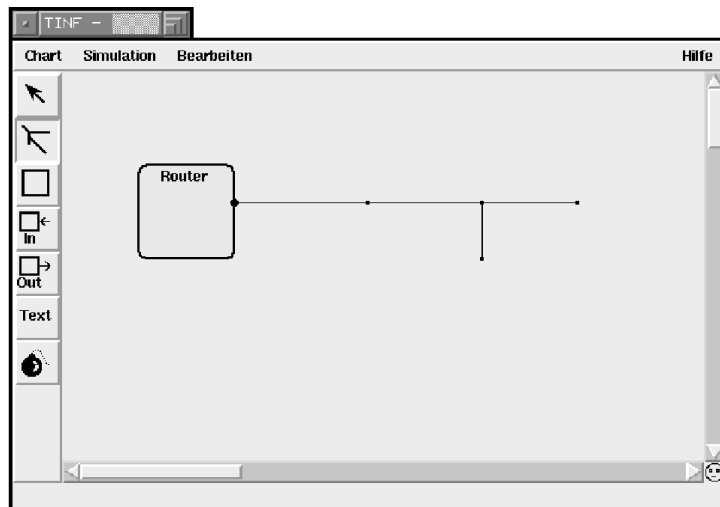


Abb. 4.9: Teilbaum wurde verschoben

4. Schritt: Löschen von Knoten

Dabei ergibt sich, daß der zweite Knoten der Leitung nun nichts mehr zur eigentlichen Struktur beiträgt und deshalb gelöscht werden kann.

Führen Sie dies durch, indem Sie den Knoten durch einen Klick markieren und dann die Bombe betätigen.

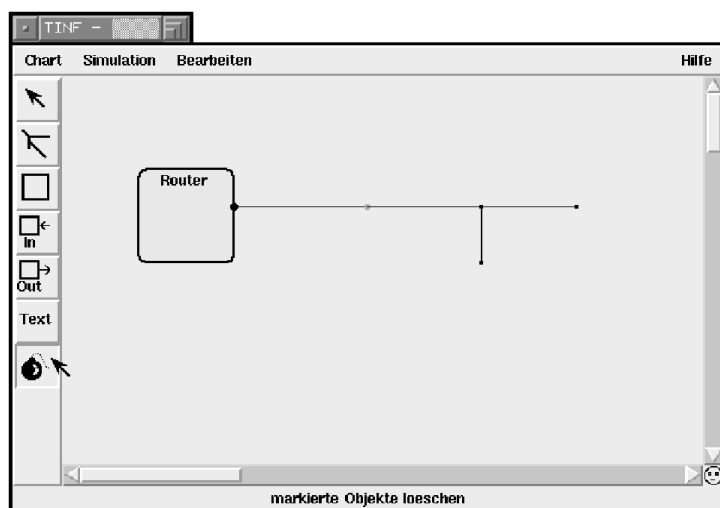


Abb. 4.10: Löschen eines Knotens

Löschen Sie nun den rechten Endknoten auf die gleiche Weise. Dabei verschwindet natürlich auch das hinführende Kabelstück.

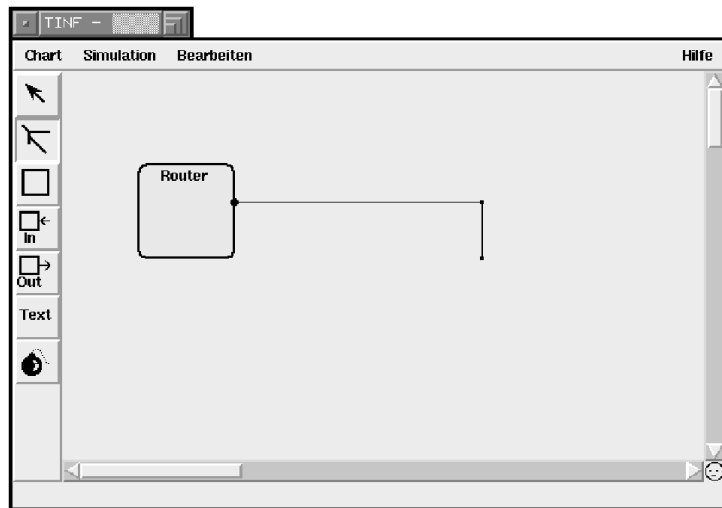


Abb. 4.11: Diagramm nach Löschen eines weiteren Knotens

5. Schritt: Erweitern des Diagramms

Wenn Sie nun eine zusätzliche Box einfügen (Modus umschalten nicht vergessen) können Sie diese sofort an das untere Ende der Leitung plazieren. Wie beim manuellen Definieren des Boxtyps erscheint auch hier eine Dialogbox zum Laden, da TINF sonst nicht weiß, über wieviel Eingänge die neue Box verfügt.

Wählen Sie hier ebenfalls einen Router aus.

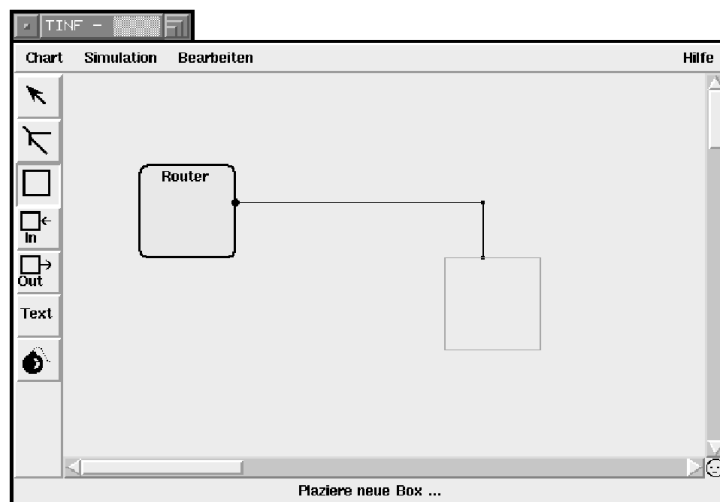


Abb. 4.12: Anfügen einer neuen Box an die Leitung

Der Router verfügt über zwei Eingänge. Wählen Sie im automatisch erscheinenden Auswahlmenü den ersten aus.

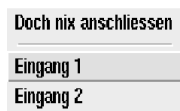


Abb. 4.13: Auswahlmenü für den anzuschließenden Eingang

Schließen sie nun eine weitere Leitung an den ersten Router an.

Dabei sehen sie, wie in dem Auswahlmenü für die Ausgänge der bereits belegte Ausgang heller gezeichnet erscheint und nicht mehr ausgewählt werden kann.

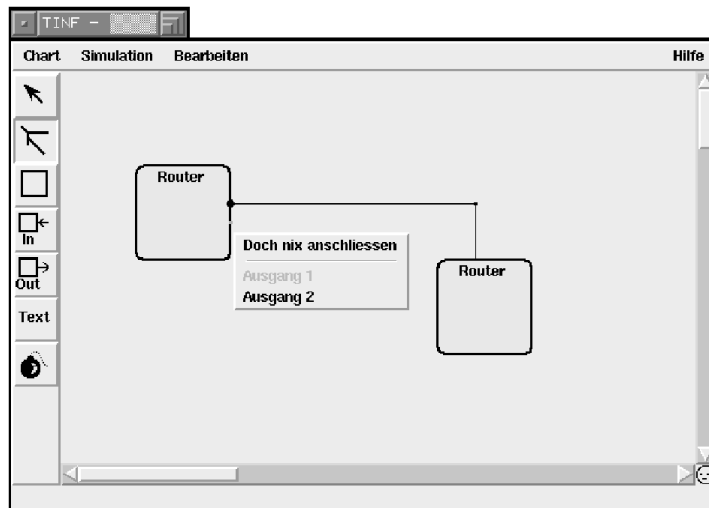


Abb. 4.14: Auswahlmenü mit belegtem Ausgang 1

Verlängern Sie anschließend die Leitung nach unten und schließen Sie einen dritten Router mit Eingang eins an.

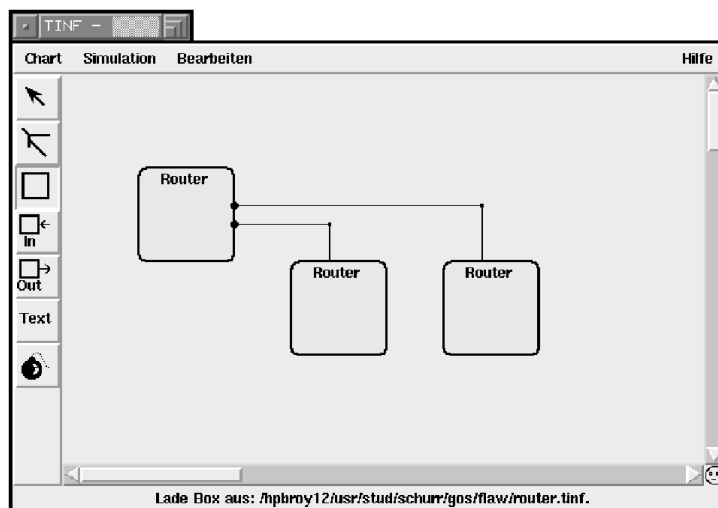


Abb. 4.15: Um dritten Router erweitertes Diagramm

6. Schritt: Markieren und Verschieben von Gruppen

Um weitere Boxen und Eingänge links von den bisherigen erstellen zu können, müssen die bestehenden zuerst verschoben werden. Dazu könnten Sie natürlich alle Objekte einzeln verschieben, würden dabei aber erneut nach den Anschlüssen gefragt.

Einfacher ist es, alle Objekte zu markieren und gemeinsam zu verschieben. Damit beim Markieren eines zweiten Objekts das erste nicht wieder demarkiert wird, halten Sie einfach die 'Shift'-Taste gedrückt. Ganze Leitungen werden durch einen Doppelklick auf den Startknoten oder auf ein Leitungsstück markiert.

In unserem Beispiel stehen aber alle zu markierenden Objekte in einer Gruppe zusammen und Sie können sie einfacher mit dem Auswahlrahmen markieren.

Wählen Sie dazu den Pfeil in der Button-Leiste. Drücken Sie dann die Maustaste in der linken oberen Ecke des Arbeitsbereichs und spannen Sie mittels Ziehen der Maus einen Rahmen auf, bis dieser die Objekte umschließt. Wenn Sie nun die Taste wieder loslassen, werden alle vollständig enthaltenen Objekte markiert.

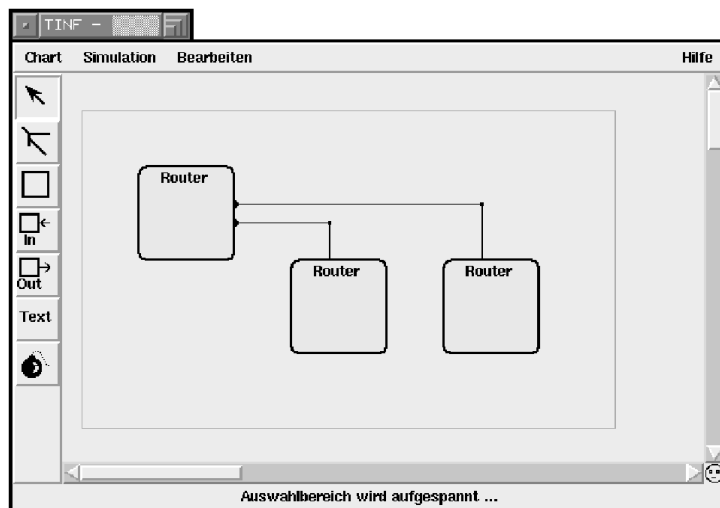


Abb. 4.16: Markieren mit dem Auswahlrahmen

Das Verschieben erfolgt analog zum Verschieben des Teilbaums, indem eines der Objekte angefaßt und dann die Maus bewegt wird. Wieder erscheinen alle Objekte dabei in einer schematischen Darstellung.

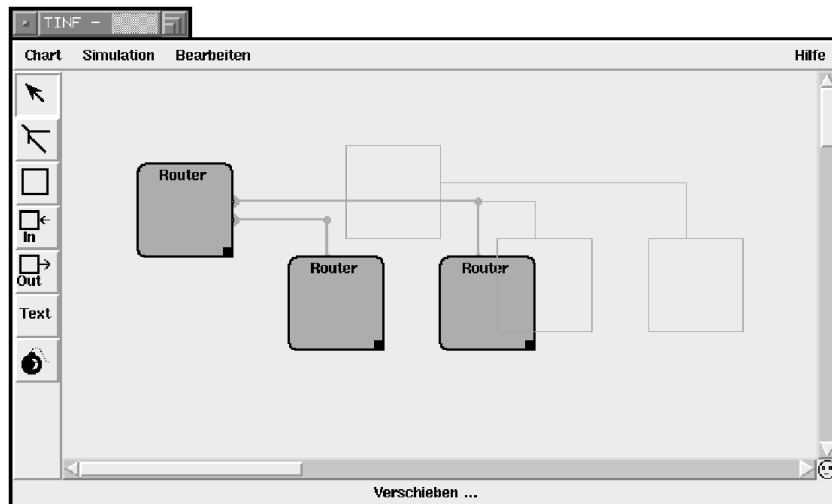


Abb. 4.17: Verschieben mehrerer Objekte

7. Schritt: Veränderung der Größe von Boxen

Erstellen Sie nun am frei gewordenen Platz eine „Once“-Box.

Sie wird oft direkt nach Eingängen geschaltet, da sie die Daten an ihrem Eingang nur einmal weitergibt und dann solange das leere Datum an ihren Ausgang legt, bis sich die Eingangsdaten wieder ändern.

Wenn die Box markiert ist, zeigt sich an der rechten, unteren Ecke ein kleines Viereck, die sogenannte Growbox. Fassen Sie diese mit der Maus an und ändern Sie dadurch die Größe, wie in Abb. 4.18 dargestellt.

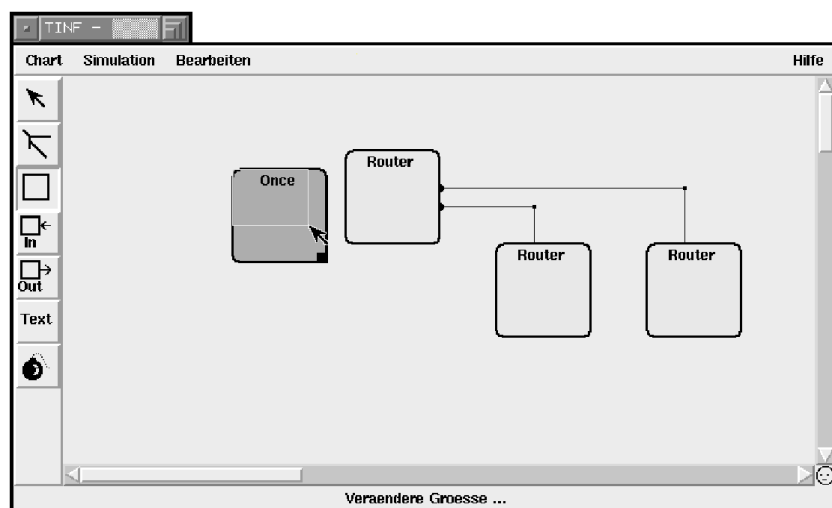


Abb. 4.18: Veränderung der Größe einer Box

8. Schritt: Vervollständigung zum fertigen Diagramm

Wenn Sie nun über den „In“-Button noch zusätzlich einen neuen Diagramm-Eingang plazieren und diesen mit der „Once“-Box verbinden, werden Sie feststellen, daß dabei kein Auswahl -
menü erscheint, da Once nur einen und der Eingang unendlich viele Anschlüsse bereitstellt.

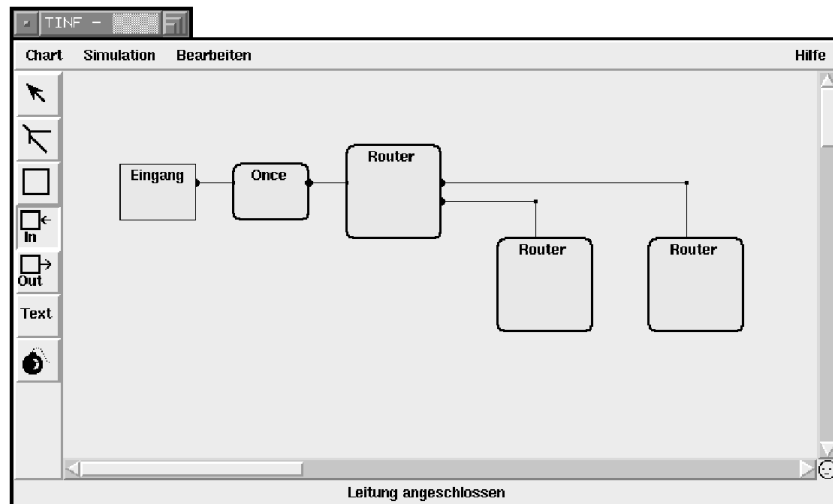


Abb. 4.19: Eingang ist mit „Once“ verbunden

Um aus diesen Objekten nun ein fertiges, sinnvolles Diagramm zur Simulation zu erstellen sind noch mehrere, zu den obigen analoge Schritte nötig.

Laden Sie nun zur Vereinfachung das fertige Diagramm über das Menü „Chart/Laden...“ aus der Datei „Tut_fertig.tinf“. Der Dialog zur Dateiauswahl entspricht dem in Abb. 4.3 dargestellten.

Dargestellt wird eine Weiche, die Daten von den vier Eingängen zu den vier Ausgängen weiterleitet. Damit die Ausgabe für mehrere Simulationsschritte sichtbar bleibt, ist vor den Ausgängen ein „Hold“-Speicher geschaltet, der ein leeres Datum an seinem Eingang ignoriert und den alten Wert weiterhin an den Ausgang legt.

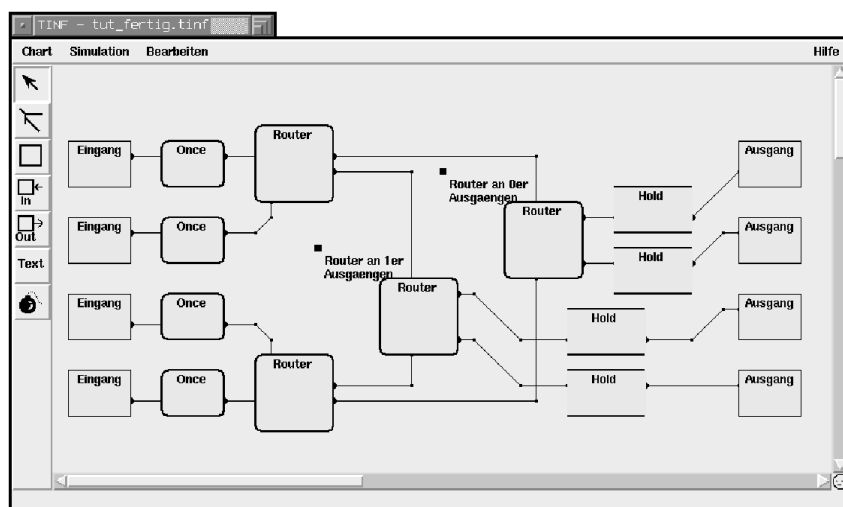


Abb. 4.20: Das Diagramm „Tut_fertig.tinf“

Die Daten am Eingang können als Präfix die Zahlen '0' oder '1' haben. Trifft das Datum auf einen Router, so stellt er je nach Wert seine „innere Weiche“ auf den entsprechenden Ausgang und gibt das Datum dort mit abgeschnittener ersten Zahl weiter.

Durch das Hintereinanderschalten zweier Router können also die vier Ausgänge über die Präfixe '00', '01', '10' und '11' angesteuert werden.

Die Router merken sich die Weichenstellung auch für Daten ohne ein solches Präfix.

9. Schritt: Simulation

Führen Sie nun im Menü „Simulation“ einen „Schritt“ aus.

Dabei erfolgt noch keine Simulation, sondern es werden alle Boxen initialisiert und die Daten ausgegeben. Von den Daten auf den Leitungen werden nur die ersten Zeichen dargestellt, um die Darstellung übersichtlich zu halten.

Die Boxen bestimmen selbst, was in ihrem Innern dargestellt wird.

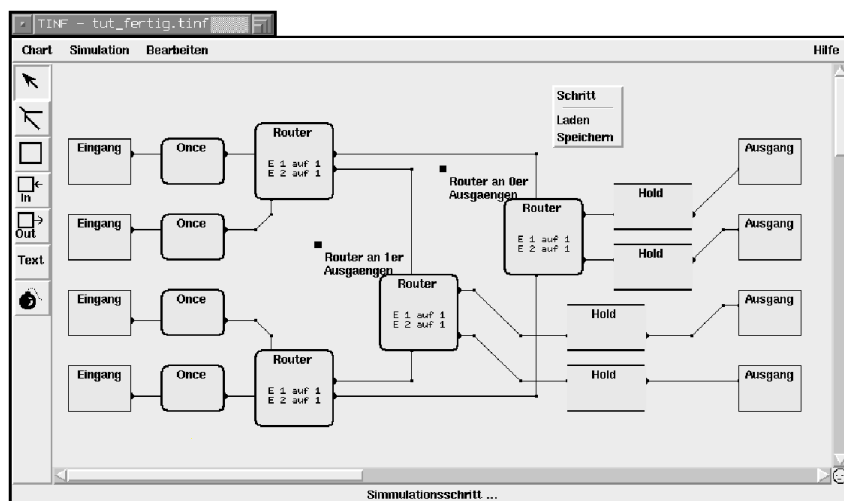


Abb. 4.21: Initialisierung der Simulation durch einen „Schritt“

Um das Durchführen mehrerer Simulationsschritte zu vereinfachen, ziehen Sie einfach das Menü „Simulation“ mit der mittleren Maustaste auf die Arbeitsfläche und betätigen Sie mehrmals die Auswahl „Schritt“.

Sie können die Simulation beeinflussen, indem Sie die Daten der Eingänge oder zu Testzwecken auch direkt die Daten auf den Leitungen ändern.

Dies geschieht über das Kontextmenü der Eingänge bzw. der Leitungen.



Abb. 4.22: Ändern der Daten über das Kontextmenü

In der folgenden Abbildung 4.23 wurde der Text mit mehreren Simulationsschritten zum Ausgang '01' geschickt.

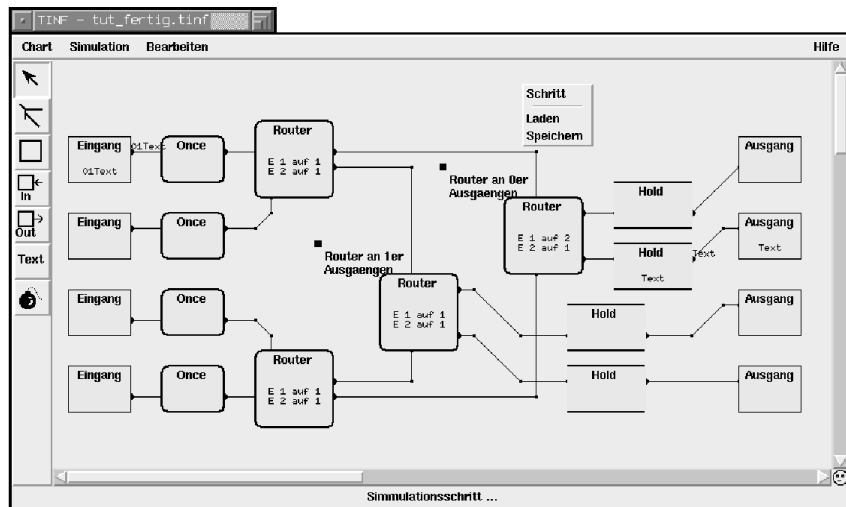


Abb. 4.23: Diagramm nach einigen Simulationsschritten mit Eingabedaten

Sie sollten nun in der Lage sein selbständig Diagramme und Simulationen zu entwerfen. Bei Bedarf können Sie auch einzelne Arbeitsschritte im Kapitel 5 „Das Arbeiten mit TINF“ nachschlagen, insbesondere das Erstellen eigener Prozesse in der Datei „SIMBAR.gos“ und der zugehörigen Definitionsdateien der Box.

5. Das Arbeiten mit TINF

Zum Starten des Programms muß das Shellskript 'tinf' ausgeführt werden. Das Skript startet automatisch den GOS-Interpreter mit dem benötigten Sourcecode. Der Pfad TINFDIR für den GOS-Interpreter muß (falls noch nicht geschehen) richtig gesetzt sein. ACHTUNG: Die benützte Workstation sollte mind. 60 MB Hauptspeicher haben, da sonst ein Arbeiten mit TINF fast nicht möglich ist. Das Speicherlimit für Prozesse muß entsprechend auf einen Wert über 90 Megabyte eingestellt sein.

Nach dem Starten erscheint ein Fenster mit einer leeren Arbeitsfläche (Abb. 5.1), an deren rechten und unteren Rand sich Scrollbalken befinden. Am linken Rand befindet sich die Funktionsleiste mit den zur Auswahl stehenden Werkzeugen. Am unteren Rand befindet sich eine Informationsleiste, in der je nach Arbeitsvorgang Hinweise zur Handhabung des Programms gegeben werden. Am oberen Rand befindet sich eine Menüleiste, mit der alle wichtigen Befehle ausgeführt werden können.

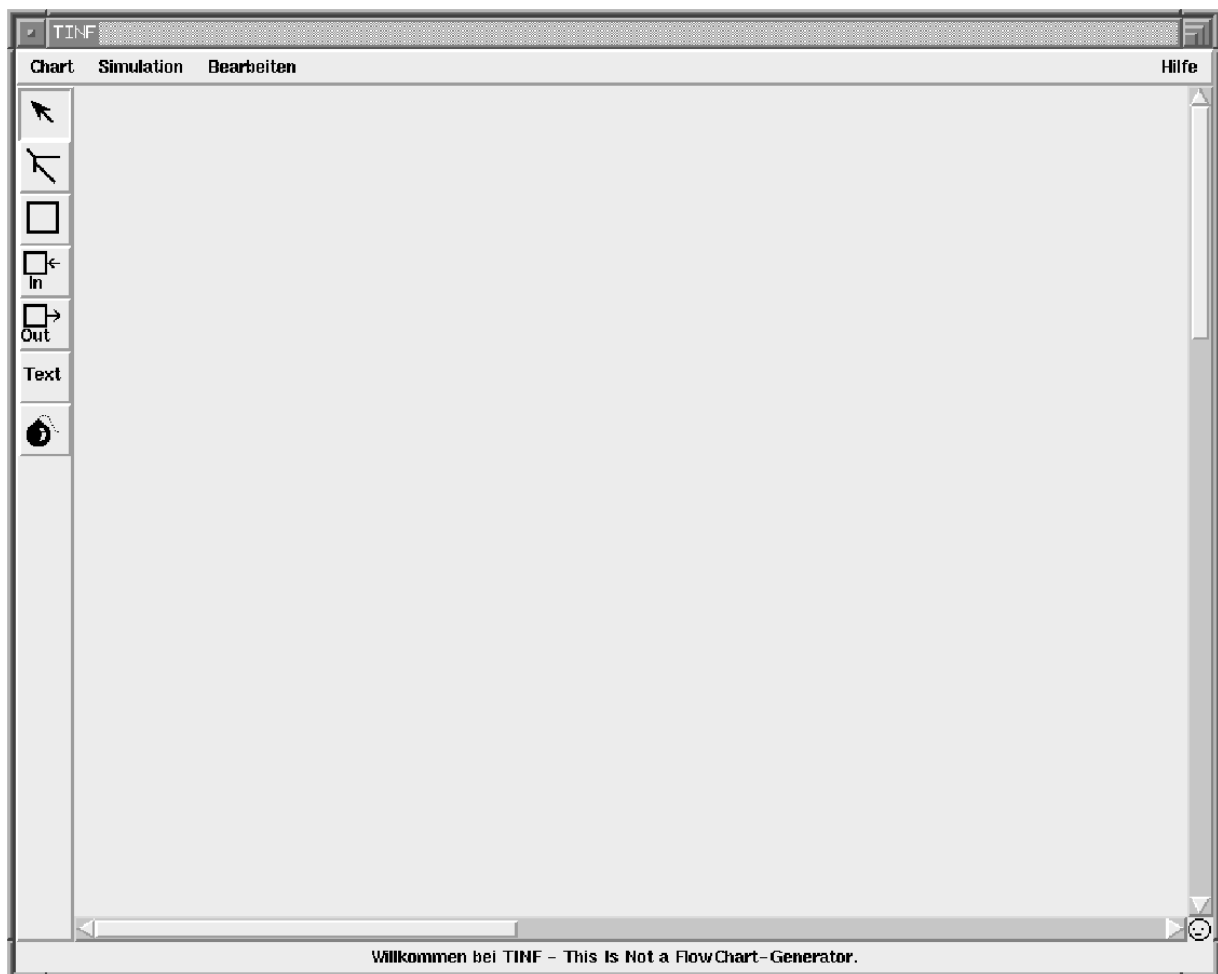


Abb. 5.1: TINF nach dem Programmstart

5.1 Editor

Als erstes wird eine Einführung in den Editor gegeben.

Bei der Konzeption der Benutzerschnittstelle wurde darauf geachtet, daß diese so weit wie möglich an bestehende Standards angepaßt ist, so daß es kaum einer Umgewöhnung von anderen Programmen bedarf.

Neue Objekte verlegen:

Zum Neuerstellung von Objekten muß zuerst die Objektart an der Funktionsleiste eingestellt werden. Es stehen Leitungen, Boxen, Eingänge, Ausgänge und Beschreibungstexte zur Auswahl. Die Buttons für die fünf verschiedenen Typen werden jeweils durch ein Viereck, einen angedeuteten Kabelbaum, ein Rechteck mit rein- bzw. herausführendem Pfeil und durch „Text“ dargestellt.

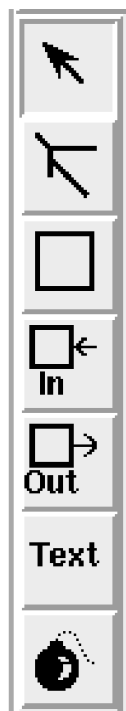


Abb. 5.2: Funktionsleiste

Im folgenden werden die Ein- und Ausgänge als Box behandelt, da es nur eine Unterscheidung in der Funktionalität gibt, die Objekte sich ansonsten aber gleich bearbeiten lassen. Der Pfeil ist ein allgemeines Auswahlwerkzeug, zum Anwählen der verschiedenen Objekte. Die Bombe dient zum Löschen bereits platzierter Objekte.

Nachdem der richtige Objekttyp ausgewählt wurde, kann mit der linken Maustaste ein neues Objekt erzeugt und platziert werden. Dabei ist zu beachten, daß die Maustaste über einem freien Feld der Arbeitsfläche gedrückt wird. Je nach Objekttyp erscheint dann eine verschiebbare

Box, ein Anfangspunkt einer neuen Leitung bzw. eine kleine Box um das jeweilige Objekt zu plazieren. Dies geschieht durch das Loslassen der Maustaste über einem freien Feld.

Das Programm prüft automatisch, ob die Plazierung zulässig war. Werden Konflikte mit bereits vorhandenen Objekten festgestellt, so kann die Neuverlegung nicht durchgeführt werden. Andernfalls erscheint bei Plazierung eines Textes ein Dialogfeld um den Text einzugeben. Der Text selbst wird nicht auf Überlappung getestet, nur sein Anfasspunkt. Ein nachträgliches Ändern läßt sich durch einen Doppelklick mit der linken Maustaste auf den Anwählknopf (siehe Abb. 5.3) des Beschreibungstextes in der linken oberen Ecke ausführen.

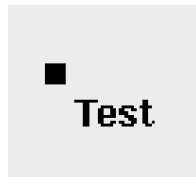


Abb. 5.3: Beschreibungstext mit Anfasspunkt

Eine neu plazierte Box ist standardmäßig undefiniert. Die Verbindung mit einer Definitionsdatei erfolgt über das Kontextmenü, aktiviert mit der rechten Maustaste (vgl. Abb. 5.4) bzw. bei Anschluß des ersten Kabels. In der Definitionsdatei stehen Informationen über die Box, die TINF benötigt um Kabel mit ihr verbinden zu können. Die Definitionsdatei einer Box besitzt das Suffix „.tinf“.

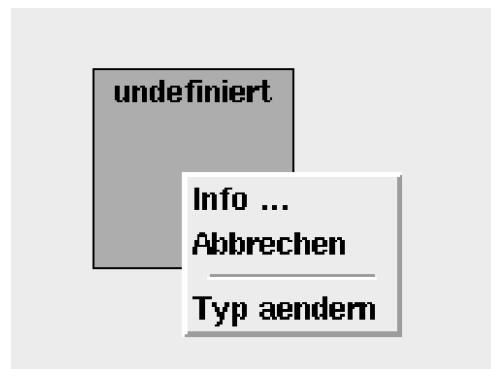


Abb. 5.4: Kontextmenü einer undefinierten Box



Abb. 5.5: Dateialogbox

Leitungen können erweitert werden, indem bei aktivem Leitungswerkzeug mit der linken Maustaste auf das unmarkierte Kabel geklickt und ein neues Leitungsstück gezogen wird. Falls der Klick auf ein Kabelstück trifft, wird an dieser Stelle automatisch ein neuer Knoten eingefügt.

Objekte anwählen/ dazuwählen: Objekte auf der Arbeitsfläche können durch einen Klick mit der linken Maustaste angewählt werden. Dabei ist egal welches Werkzeug im Moment aktiv ist. Das ausgewählte Objekt erhält eine Markierungsfarbe und alle anderen Objekte werden deaktiviert.

Beschreibungstexte werden angewählt, indem auf den Anfasspunkt (siehe Abb. 5.3) in der linken oberen Ecke geklickt wird.

Bei den **Leitungen** gibt es verschiedene Möglichkeiten des Anwählens: Es können Knoten, Kabelstücke, Kabelbäume oder die ganze Leitung angewählt werden.

Zum Anwählen eines Knotens oder eines Kabelstückes klickt der Benutzer mit der linken Maustaste auf die entsprechende Komponente. Um einen Kabelbaum anzuwählen, muß ein Doppelklick auf den Knoten erfolgen, ab dem der Baum markiert sein soll. Die ganze Leitung wird aktiviert indem der Benutzer entweder einen Doppelklick auf den Startknoten oder ein Kabelstück ausführt.

Die **Boxobjekte** werden durch einen Klick auf die jeweilige Box aktiviert.

Das Dazuwählen erfolgt vollkommen analog. Zusätzlich muß beim Anwählen die ‘Shift’-(Hochstell)-Taste gedrückt sein. Damit wird verhindert, daß bereits angewählte Objekte deaktiviert werden. Beim Dazuwählen können allerdings nur „kompatible” Markierungszustände zusammen angewählt werden: Knoten und Kabelstücke werden nie zusammen mit anderen Objekten markiert, nur einzeln.

Die Möglichkeit ganze Gruppen von Objekten an- und dazuzuwählen bietet das **Auswahlwerkzeug** (der oberste Knopf auf der Funktionsleiste). Ein Klick auf ein freies Feld auf der Arbeitsfläche öffnet einen Rahmen, der über die vorhandenen Objekte gezogen werden kann. Jedes Objekt, welches sich beim Loslassen der Maustaste innerhalb dieses Rahmens befindet, wird aktiviert (s. Abb. 5.6). Falls zusätzlich die ‘Shift’-Taste gedrückt wird, erfolgt wiederum ein Dazuwählen. Zum Deaktivieren angewählter Objekte muß mit dem aktivem Auswahlwerkzeug (Pfeil) auf einen freien Teil der Arbeitsfläche geklickt werden. Einzelne markierte Objekte lassen sich auch durch erneutes Anwählen mit Shift wieder demarkiert.

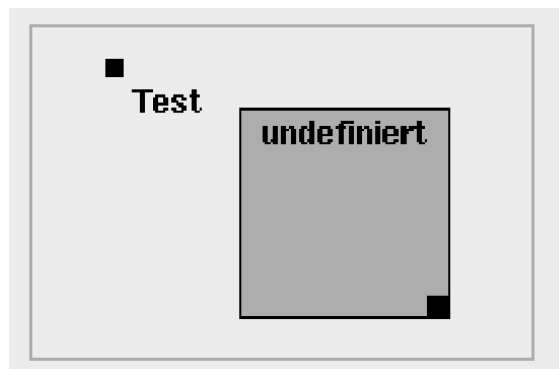


Abb. 5.6: Aktivieren der Objekte mit dem Auswahlwerkzeug

Verschieben/ Größe ändern:

Zum **Verschieben** müssen die Objekte zuerst angewählt werden. Dabei können beliebige Objektgruppen gleichzeitig verschoben werden. Der Benutzer klickt auf ein bereits angewähltes Objekt und „zieht” es zu seiner neuen Position. Dort läßt er die Maus taste los und TINF versucht dieses und alle mitangewählten Objekte um diese Distanz zu verschieben. Dabei gelten die gleichen Regeln wie bei der Neuplazierung.

Beim Verschieben von Teilbäumen werden die Verbindungen getrennt. Falls der Startknoten des abgetrennten Teilbaums auf einer Diagonalen bzw. Geraden, vom ursprünglichen Startknoten aus gesehen, zum Liegen kommt (Abb. 5.7), wird versucht die Verbindung über diese Diagonale bzw. Gerade neu zu legen. Ist dies nicht möglich und durch das Abtrennen würden einzelne Knoten stehen bleiben, so werden diese Knoten entfernt.

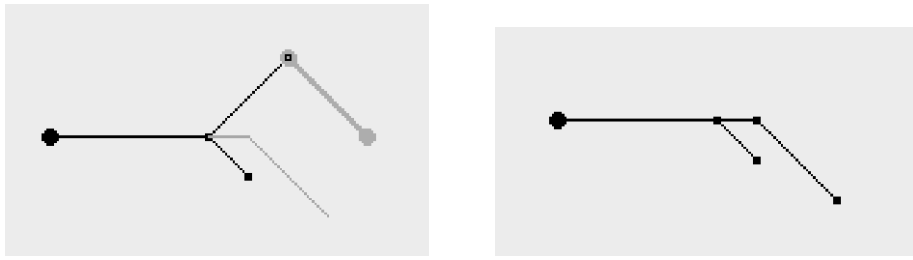


Abb. 5.7: Startknoten des Teilbaums kommt beim Verschieben auf einer Diagonalen zu liegen

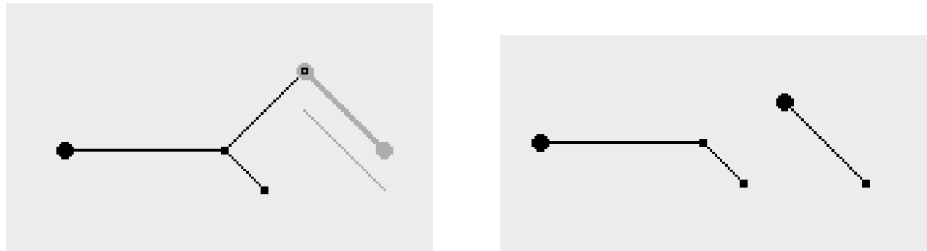


Abb. 5.8: Teilbaum wird beim Verschieben abgetrennt

Zum **Größe ändern** der Box muß sie als einziges Objekt angewählt sein. Nun erfolgt der Klick mit der linken Maustaste in die Growbox der Box (das kleine schwarze Rechteck am unteren rechten Rand). Der Benutzer zieht das Objekt in die vorgesehene Größe und läßt die Maustaste los (siehe Abb. 5.9). Dabei ist zu beachten, daß die Box nicht kleiner gemacht werden kann, als in der zugehörigen Definitionsdatei vorgeschrieben.

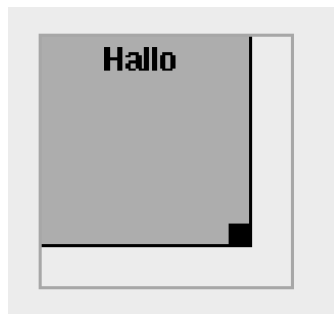


Abb. 5.9: Größe ändern

Anschließen:

Zum Anschließen und Verbinden von Boxen gibt es zwei Möglichkeiten. Zum einen kann direkt auf dem Boxrand eine neue Leitung verlegt werden und zum anderen können Kabelenden und Startknoten auf Boxränder geschoben werden oder umgekehrt. In beiden Fällen erscheint unmittelbar darauf ein Menü (vgl. Abb. 5.10), in dem gefragt wird, mit welchem Ein- bzw. Ausgang der Knoten verbunden werden soll. Der Benutzer sieht dabei auch gleich, wieviel freie Anschlußmöglichkeiten die jeweilige Box noch hat. Falls die Box nur eine einzige oder eine unbeschränkte Anzahl an Ein- und Ausgängen zuläßt, erscheint dieses Menü nicht.



Abb. 5.10: Menü mit Anschlußmöglichkeiten

Beim Verbinden von zwei Kabeln muß darauf geachtet werden, daß Knoten nur mit den etwas dickeren Startknoten angeschlossen werden können und umgekehrt. Falls eine Leitung an ein Leitungsstück angeschlossen werden soll, muß vorher mit dem Leitungswerkzeug ein Knoten an der geplanten Verzweigung erzeugt werden.

Undo/ Redo.

TINF hat eine eingebaute Undo/ Redo-Funktion, die unter dem Menü Bearbeiten (Abb. 5.11) zu finden ist. Mit ihr können die zuletzt durchgeführten Aktionen rückgängig gemacht (Undo) bzw. die rückgängig gemachten Aktionen wieder ausgeführt (Redo) werden. Die Undo-Funktion kann sich beliebig lange Funktionsfolgen merken, d.h. daß die Ausgangsposition immer wieder erreicht werden kann. Die Redo-Funktion merkt sich (im Prinzip) auch beliebig lange Undofunktionsaufrufe. Allerdings wird der Redospeicher gelöscht, sobald ein Element neu platziert oder in irgendeiner Weise verändert wurde. Dies ist nötig, da nach einer Änderung, unter Umständen kein korrektes Redo mehr durchgeführt werden kann, da sich die Positionen der Objekte geändert haben könnten.

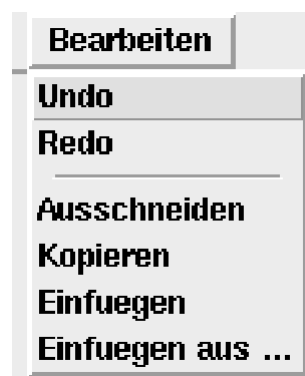


Abb. 5.11: Menü Bearbeiten

Cut, Copy & Paste:

Ebenfalls unter dem Menü Bearbeiten (Abb. 5.11) befinden sich die Ausschneide-, Kopier- und Einsetzfunktionen. Diese ermöglichen das Duplizieren von beliebigen Objekten auf der

Arbeitsfläche und das Kopieren von Dokument zu Dokument. Es ist darauf zu achten, daß nur ganze Objekte kopiert und ausgeschnitten werden können. Es ist also nicht möglich, einzelne Knoten, Kabelstücke oder Teilbäume zu kopieren oder auszuschneiden.

Nach dem Anwählen der zu duplizierenden Objekte wird der entsprechende Befehl aufgerufen und dabei die aktivierten Objekte auf die Festplatte gespeichert. Beim Befehl „Aus schneiden“ werden die Objekte zusätzlich von der Arbeitsfläche entfernt. Ein „Einfügen“ kann jederzeit erfolgen. Dabei werden die auf der Festplatte gespeicherten Objekte geladen und können vom Benutzer plaziert werden. TINF überprüft wiederum auf Überlappung und verwirft gegebenenfalls die Aktion.

Löschen:

Im Unterschied zum Ausschneiden können mit dem Befehl Löschen (die Bombe in der Funktionsleiste (siehe Abb. 5.2) auch einzelne Endknoten (Abb. 5.13), Knoten zwischen Kabelstücken ohne Verzweigung und Knick und einzelne Kabelstücke gelöscht werden. Dabei wird keine Kopie der Objekte auf der Platte angefertigt.

Beim Entfernen von Kabelstücken entstehen normalerweise zwei neue Leitungen (siehe Abb. 5.12), wenn nicht zufällig einzelne Knoten stehenbleiben, die gelöscht werden müssen.

Knoten auf geraden Leitungszügen (ohne Verzweigung) verschwinden beim Löschen ohne weitere Auswirkungen.



Abb. 5.12: Vor und nach dem Entfernen eines Kabelstückes



Abb. 5.13: Löschen eines Endknotens

Speichern/ Laden:

Natürlich bietet TINF auch die Möglichkeit zum Speichern und Laden fertiger Datenflußdiagramme. Diese Befehle befinden sich unter dem Menüpunkt Chart (Abb. 5.14).

Beim Laden wird das vorher erstellte Datenflußdiagramm gelöscht! Mit dem Befehl „Speichern unter...“ kann das Datenflußdiagramm mit einem neuen Namen abgespeichert werden, dabei bleibt die alte Datei erhalten. Datenflußdiagramme werden mit dem Suffix ' .tinf' abge

speichert. Es besteht zudem die Möglichkeit das Datenflußdiagramm als PostScript®-Datei abzuspeichern. Dabei wird immer die ganze Grafik, nicht nur das Markierte, abgespeichert.



Abb. 5.14: Das Chart-Menü

5.2 Simulator

Da der Editor mit dem Simulator in einem Programm vereint ist, kann jederzeit mit der Simulation begonnen werden. Dadurch ist es möglich sich langsam an sein gewünschtes Ergebnis heranzuarbeiten und die Arbeit durch regelmäßige Simulationstestläufe zu überprüfen.

Werte setzen:

Die Eingabe der Simulationsdaten kann auf vielfältige Weise vorgenommen werden. Als erste Möglichkeit können direkt die Daten auf den Leitungen manipuliert werden. Dies geschieht durch einen Klick mit der rechten Maustaste auf eine Leitung. Es erscheint das Kontextmenü der Leitung mit einem Menüpunkt zum Verändern der Leitungsdaten (siehe Abb. 5.15).

Nachdem der Befehl „Daten ändern“ gewählt wurde, erscheint ein Dialogfeld, das die aktuellen Daten der Leitung anzeigt. Dieses Feld kann beliebig modifiziert werden und die Leitung übernimmt für den nächsten Simulationsschritt (vgl. Abschnitt über das Simulieren) die eingegebenen Daten.



Abb. 5.15: Das Kontextmenü der Leitungen

Als zweite Möglichkeit bieten sich die Eingänge (siehe Objekte verlegen) an. Durch einen Klick mit der rechten Maustaste auf ein Eingangsobjekt erscheint, analog zu den Leitungen,

ein Kontextmenü (Abb. 5.16) mit dem Befehl „Eingabedaten ändern“. Nachdem dieser angewählt wurde, können die Daten der Box verändert werden. Der Eingang gibt bei jedem Simulationsschritt diesen Wert an eine angeschlossene Leitungen weiter.

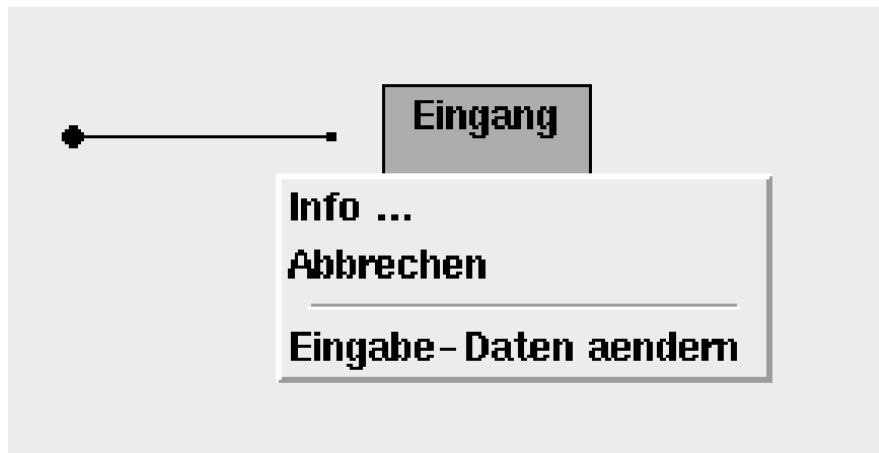


Abb. 5.16: Kontextmenü der Eingänge

Simulieren:

Die Simulation wird gestartet, indem im Menü „Simulieren“ der Befehl „Simulationsschritt“ gegeben wird. Es kann immer nur ein Simulationsschritt auf einmal ausgeführt werden.

Die Boxen haben ein spezielles Textfeld, in das sie Informationen über ihren inneren Zustand hineinschreiben können. Zur Anzeige der Simulationsergebnisse eignen sich hier die Ausgangsboxen (vgl. Objekte verlegen) und die vordefinierten „hold“ Boxen. Sie zeigen beide den Inhalt der angeschlossenen Leitung an.

Während die Ausgangsbox bei jedem Simulationsschritt den Wert der darauf hinführenden Leitung übernimmt (auch das leere Datum), speichert der „hold“-Prozeß den letzten Wert solange bis neue Daten anliegen.

Implementierung der SIMBARs:

Zur Erzeugung neuer simulierbarer Objekte (SIMBARs), muß der Benutzer eine neue GOS-Klasse schreiben, die er in der Sourcedatei „SIMBAR.gos“ hinzufügt. Diese Klasse muß Unterklasse von (SUBCLASSOF) SIMBAR sein, damit die vorgeschriebene Schnittstelle (vgl. Bedienkonzept) benützt werden kann. Die Methoden können überladen und somit ein völlig neuer Datenflußprozeß oder Speichertyp kreiert werden.

Damit die neudefinierten Boxen vom Editor mit den grafischen Boxen auf der Arbeitsfläche verbunden werden können, sind sie zusätzlich im SIMBAR-Generator einzubinden. Die Klasse GETSIMBARs (der SIMBAR-Generator) befindet sich ebenfalls in der Sourcedatei „SIMBAR.gos“. Im Grunde handelt es sich dabei um eine große IF...ELIF-Abfrage. Hier wird der Name der Box eingetragen, die mit der neuen SIMBAR Unterklasse verbunden werden soll:

```
...
ELIF s == "ownbox" THEN           -- Name der neuen Box
    ob := OWNBOX!NEW(...);       -- neue Unterklasse
    sb := coerceSIMBAR ob;
ELIF s == ...
```

Eine andere, einfache Art und Weise neue Boxobjekte zu erschaffen bietet sich durch die vordefinierte Klasse SIMPLESIMBAR. Für diese Klasse muß nur eine Gofer-Funktion angegeben werden. Die Gofer-Funktion legt die Abbildung der Eingänge auf die Ausgänge fest und ist vom Typ [String] -> [String].

Daten werden durch Strings dargestellt und die verschiedenen Ein- bzw. Ausgänge durch Stringlisten, wobei die Kardinalität der Listen die Anzahl der jeweiligen Ein- bzw. Ausgänge festlegt:

```
...
simplefunc :: [String] -> [String]
simplefunc sl = [ sl!!0, sl!!1, concall sl ] -- Weiterreichen der
                                           -- ersten beiden Werte
                                           -- und konkatenieren
                                           -- aller Eingänge auf
                                           -- Ausgang 3
...
ELIF s == "simplefunc" THEN
    sf := SIMPLESIMBAR!NEW(simplefunc);
    sb := coerceSIMBAR sf;
ELIF ...
```

Die Möglichkeit komplexere Datenflußprozesse und -speicher zu simulieren bietet die Klasse COMPLEXSIMBAR an. Sie arbeitet ähnlich wie die SIMPLESIMBAR Klasse; es gibt aber darüber hinaus die Möglichkeit einen inneren Zustand zu definieren und eine Funktion die diesen Zustand in Abhängigkeit von der Eingabe manipuliert.

Eine Instanz der Klasse COMPLEXSIMBAR erhält also zwei Goferfunktionen und einen inneren Zustand bei der Initialisierung. Bei der Ausführung der inneren und äußeren Funktion wird die äußere zuerst ausgewertet.

```
...
-- aufsummieren aller Eingabelistenelemente
sumup :: [String] -> Int
sumup (s:xs) = a2i s + sumup xs
sumup [] = 0

-- Abb. der Eingabe auf inneren Zustand
innerfunc :: [String] -> [String] -> [String]
innerfunc inlist innerdat = [( show ( sumup inlist
                                   + a2i (head innerdat) ) )]

-- Abb. der Eingabe auf die Ausgabe
```

```
outerfunc :: [String] -> [String] -> [String]
```

```
outerfunc sl _ = sl
```

```
...
```

```
    ELIF s == "Plus" THEN
```

```
        plus := COMPLEXSIMBAR!NEW(outerfunc,innerfunc,["0"]);
```

```
        sb := coerceSIMBAR plus;
```

```
    ELIF ...
```

6. Vordefinierte Objekte

Zum Arbeiten und Experimentieren enthält die Objektbibliothek eine ganze Reihe vordefinierter Objekte. Objektbibliothek heißt in diesem Fall, daß die Objekte in der Datei SIMBAR.gos implementiert sind und die zugehörigen Definitionsdateien im öffentlich zugänglichen Verzeichnis von TINF (pub/lib/) liegen.

Es folgt eine Beschreibung der Handhabung der vordefinierten Objekte. Der Leser sollte vorher das Kapitel über die Implementierung der SIMBARs gelesen haben.

Es wurden die beiden logischen Grundfunktionen **Or** und **And** implementiert. Dabei handelt es sich um n-stellige Funktionen. In der Definitionsdatei muß die Stelligkeit und der Name des SIMBARs angegeben werden.

```
-- NIn -1
-- NOut 1
-- Name And
-- Bsp. für beliebig-stelliges 'And'
```

Es existieren dazu die Definitionsdateien 'and.tinf' und 'or.tinf'. Beiden Boxen können beliebig viele Eingänge besitzen.

Zusätzlich zu den beiden Grundfunktionen gibt es ein vordefiniertes **Not** und ein **Xor** Gatter. Zur Implementierung weiterer bool'scher Funktion kann die Klasse COMPLEXSIMBAR benutzt werden, wobei als innere Funktion die Gofershilfsfunktion *csbBoolFktInner* und als äußere Funktion *csbBoolFktOuter* mit der gewünschte Funktion (*[Bool] -> Bool*) als Parameter übergeben wird.

Alle bool'schen Funktionen puffern Eingaben mit einer Länge größer eins und Werten sie Stück für Stück aus.

```
...
halbadd :: [Bool] -> Bool
...
    ELIF s == "Halbaddierer" THEN
        csb := COMPLEXSIMBAR!NEW((csbBoolFktouter halbadd),
                                csbBoolFktInner,
                                []::[String]);
        sb := coerceSIMBAR csb;
    ELIF ...
...
```

Die Objekte **Hold** und **Once** dienen zur Unterstützung der Aus- und Eingabeboxen.

Die Hold Box speichert den zuletzt angelegten Wert. Er wird am Bildschirm ausgegeben und liegt solange am Ausgang der Box an bis am Eingang ein nicht leeres Datum erscheint.

Die Once Box verhindert, daß ständig der gleiche Wert auf einer Leitung liegt, d.h. sie „merkt“ sich den zuletzt anliegenden Wert und läßt nur ein geändertes Datum durch.

Das Objekt **PC** simuliert einen „einfachen Netzwerk-PC“. Es hat vier Eingänge und vier Ausgänge. Sie dienen dazu, mehrere PCs miteinander zu verbinden, wobei immer Eingang j und Ausgang j mit dem selben PC verbunden sein muß. Die ersten drei Ein-/Ausgänge bieten die Möglichkeit je einen PC anzuschließen, der vierte Ausgang bedient ein „Sichtgerät“ und der vierte Eingang eine „Tastatur“ (etwa in Form einer Hold- und einer Once-Box).

Beim Simulationsstart erhält jeder PC eine individuelle Adresse (adr0...adrN). Jetzt können Nachrichten von PC zu PC verschickt werden, wobei die einzelnen Rechner nach und nach die günstigste Route für die Nachricht finden. Der SIMBAR-Generator benützt dabei das Attribut *uniqueNr* um eindeutige Adressen zu erzeugen (bzw. die Rechner durchnummerieren).

Der **Mixer** puffert seine Eingaben und mischt die anliegenden Daten. Er liest seinen Puffer aus und nimmt jeweils die ersten Zeichen und hängt sie zu einem neuen String zusammen. Diesen gibt er auf den Ausgang aus.

Der **Router** lenkt seine Daten gesteuert an die Ausgänge. Eine ‘0’ als erstes Zeichen des Eingabedatums an einem Eingang bewirkt, daß alle nachfolgenden Eingaben an diesem Eingang auf Ausgang 1 umgeleitet werden, bis einmal eine ‘1’ als erstes Zeichen anliegt. In diesem Fall werden die Daten zu Ausgang 2 geleitet. Beide Kanäle arbeiten unabhängig voneinander. Bei einer Kollision der Ausgaben (beide Eingänge liegen auf dem selben Ausgang), hat Eingang 1 Priorität der andere wird gepuffert.