



Willy Jünger\*, Jürg Mathys\* Hansueli Wettach\*\*

# Effizienz und Transparenz der EDV

Die EDV dient normalerweise dem Benutzer zur Rationalisierung oder Bereitstellung von Führungsinformationen. Um den hohen Anforderungen der achtziger Jahre genügen zu können, ist in der EDV selbst eine saubere technische Infrastruktur unerlässlich. Diese lässt sich mit vertretbarem Aufwand erstellen und macht sich durch eine bessere Effizienz und Transparenz rasch bezahlt. Zurzeit werden weitere praxisbezogene Erfahrungen gesammelt.

## 1. Aufgabe

### 1.1. Gegebenheiten

Das ausgezeichnete Preis-Leistungs-Verhältnis moderner EDV-Hardware lässt mittlere Gesellschaften in Anwendungsbereichen vorstossen, die bis anhin ausschliesslich Grossbenutzern vorbehalten waren. Hinzu kommt, dass die Lieferanten bei der Ankündigung neuer Produkte die Erwartungen der Benutzer immer höher schrauben. So ist es selbstverständlich geworden, von DB/DC, TP, On-line, Data dictionary und dergleichen mehr zu sprechen, obwohl hinter jedem dieser Schlagworte eine Fülle von Möglichkeiten – gleichzeitig aber auch von Problemen – steckt. Zwei dieser Probleme sind:

- Wie lässt sich die moderne, leistungsfähige Hardware und Software ausnutzen, als technische Infrastruktur für unsere DB/DC-Anwendungen sowie zur Rationalisierung der Entwicklungstätigkeit?
- Welche Methoden und Verfahren ermöglichen, gezielt benutzerfreundliche On-line-Anwendungen zu entwickeln, im Gegensatz zur Trial- und Error-Methode?

Soll die EDV ihre Dienstleistungen auf

Jahre hinaus zweckmässig und wirtschaftlich erbringen, so kommt sie nicht darum herum, eine geeignete Organisation und technische Infrastruktur aufzubauen. Dabei muss Klarheit über das gesamte zu entwickelnde System erlangt werden; es geht nicht an, einzelne Aspekte wie die Richtlinien, Dokumentation oder Datensicherung auf später zu verschieben. Die verschiedenen Gegebenheiten sind dermassen miteinander verknüpft, dass schon bald die Weichen gestellt sind. Der folgende Beitrag zeigt, wie wir diese Probleme mit der in Bild 1 genannten Hardware und Software gelöst haben.

### 1.2. Unsere Lösung

Die wesentlichen Merkmale unserer Lösung sind:

- Die Entwicklungstätigkeit ist weitgehend von technischem Ballast entlastet
- Analyse und Programmierung sind durch normierte Vorgaben aufeinander abgestimmt
- Die Programmierung erfolgt im Dialog, unter Verwendung vorcodierter, katalogisierter Rahmenprogramme, sowohl für die Stapelverarbeitung wie auch für On-line-Transaktionen
- Zwischen der programmierten applikatorischen Verarbeitung sowie dem Zugriff und der Verwaltung von Daten besteht eine klare Trennung, so dass eine gewisse Datenunabhängigkeit gegeben ist
- Die EDV-Dokumentation wird projektbegleitend im Dialog geführt, im Sinne der aktenarmen Sachbearbei-

tung. Auch die Konsultation erfolgt am Bildschirm, je nach Bedarf als Beschreibung, Verzeichnis oder Verwendungsnachweis

- Wir verwenden standardmässige IBM-Software, im Rahmen eigener organisatorischer Massnahmen, die mit vertretbarem Aufwand getroffen werden konnten
- Die Zugriffsberechtigung auf Daten wird in angemessener Weise überwach-

## 2. Normierte Analyse und Programmierung

Die bitteren Erfahrungen, welche bei der Wartung persönlich konzipierter und mit raffinierten Tricks versehenen Programmen gemacht werden mussten, haben Diskussionen über den Sinn der Normierung verstummen lassen. Tatsache ist, dass die Normierung dem Analytiker und Programmierer keineswegs das Denken abnimmt und ihn zum Handlanger degradiert. Sie liefert ihm bei der Programmablaufsteuerung wesentliche Hilfestellungen, indem einmal gemachte Überlegungen nicht immer wieder angestellt werden müssen. Der befehlsmässig weit umfangreichere eigentliche Verarbeitungsteil, der ja spezifisch auf die zu lösende Aufgabe ausgerichtet ist, bleibt nach wie vor zu programmieren. Dank der Normierung kann sich der Programmierer auf die einwandfreie Lösung der wirklich schwierigen applikatorischen Probleme konzentrieren.

M/

- Datenerfassung**
- Datensammelsystem Inforex 1330-I mit fünf Eingabeplässen
- Rechenzentrum**
- Zentraleinheit IBM 4331-K02, 2 Megabyte mit normaler Peripherie
  - Ersatz durch IBM 4341 im Verlauf von 1982 vorgesehen
- Teleprocessing**
- 30 Bildschirme und 12 Terminaldrucker
  - Laufender Ausbau, je nach Bedarf der Benutzer
- Software**
- Betriebssystem DOS/VSE, Release 2, DL 1, CICS, ICCF, SDF
  - Programmiersprache: Cobol

Bild 1 Hardware und Software

### 2.1. Standard-Programmlogik

Unsere Standard-Programmlogik weist folgende wesentliche Merkmale auf:

- Trennung von Steuerung und eigentlicher Verarbeitung
- Eindeutige funktionelle und hierarchische Gliederung, entsprechend den in jedem kommerziellen Programm wiederkehrenden Funktionen
- Klare Regelung der Verzweigungen im Programm
- Programmunabhängige Definition und Zugriff auf Daten

Für die Stapelverarbeitung verwenden wir seit Jahren eine Standard-Programmlogik – ähnlich SPL – gemäss Bild 2.

Die normierte Struktur für On-line-Programme ist aus Bild 3 ersichtlich. Dazu ergeben sich folgende ergänzende Erläuterungen:

- Die Transaktion wird durch Eingabe am Bildschirm oder Aufruf von einem anderen Programm gestartet
- Es bestehen Normen hinsichtlich Identifikation, Kennzeichen, Adressierung, Arbeitsspeicher, Funktionstasten, Bildschirmaufbau usw., die zwingend zu beachten sind
- Die einzelnen Transaktionsschritte sind pseudokonversationell gelöst, im Hinblick auf eine geringere Systembelastung und gute Antwortzeiten für den Benutzer
- Die DB-Update-Funktion läuft als eine eigene Task, damit Sicherstellung und Wiederanlauf gewährleistet sind

### 2.2. Rahmenvorgaben

Die normierten Programmstrukturen für Batch und On-line sind am Bildschirm in geeigneter Weise dargestellt und werden bei der Analyse verwendet. Diese kann somit auf die eigentliche Verarbeitung beschränkt werden; sie ist von jeglichem Ballast befreit.

Ein Rahmenprogramm in Cobol-Sourcecode steht in einer allgemein zugänglichen Bibliothek. Das Rahmenprogramm umfasst die Standard-Programmlogik, allgemeine Beschreibungen mit Gebrauchsanleitung usw.

Das Rahmenprogramm bildet die Basis für die individuellen Programme, das heisst:

- Es wird vom Programmierer kopiert
- Die erforderlichen Beschreibungen und Deklarationen werden ergänzt; die übrigen eliminiert
- Die eigentliche Verarbeitung wird gemäss Problemstellung aus der Analyse im Dialog programmiert

Alsdann erfolgen Umwandlungen und Tests unter Einbezug aller Komponenten, insbesondere der programmunabhängigen standardmässigen Recorddeklarationen und I/O-Modulen.

### 2.3. Erfahrungen

Die wesentlichen Vorteile normierter Programmstrukturen in Rahmenprogrammen gemäss unseren mehrjährigen Erfahrungen sind:

- Erhebliche Rationalisierung sowie geringere Fehlerquote
- Bessere Programmwartung durch die einheitliche Struktur und Namensvergebung sowie eine gewisse Personenunabhängigkeit
- Gute Kommunikation zwischen Benutzer, Analyse und Programmierung dank verständlicher Programmlogik und Transparenz

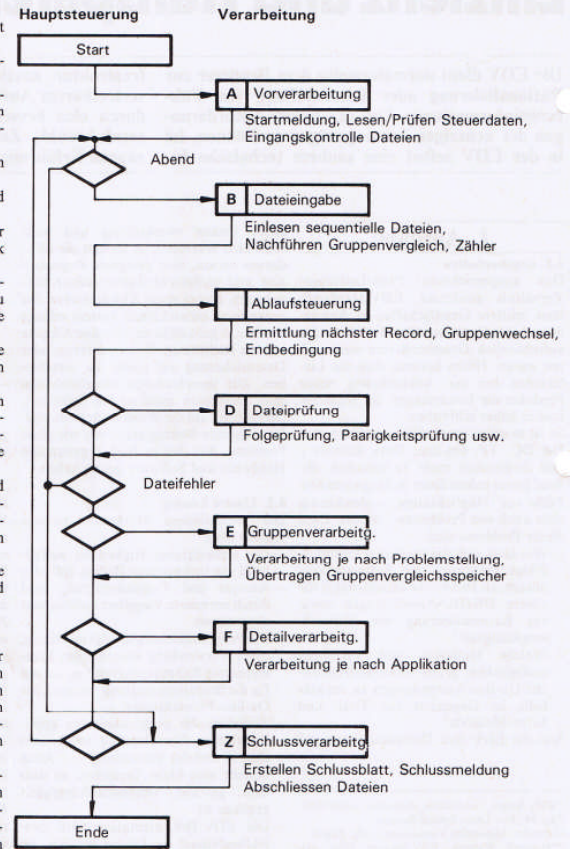


Bild 2 Standard-Programmlogik (+Nora)

### 3. Datenhandling mit Interface

Schon früh war es uns ein Anliegen, eine gewisse Datenunabhängigkeit zu erlangen. Deshalb gehen alle Zugriffe vom Programm auf den Datenbestand über ein Interface, wie in Bild 4 dargestellt. Die seit Jahren gemachten guten Erfahrungen beim Einsatz von Furegati-Modul für Eingabe und Ausgabe sequentieller Dateien haben uns veranlasst, diese Entwicklung in Richtung Datenunabhängigkeit konsequent fortzuführen.

#### 3.1. Standardmässige Recorddeklarationen

Die Recordarten von Dateien und Datenbanken, die durch mehrere Programme bearbeitet werden, sind einmalig in Cobol definiert und in der Source-Bibliothek katalogisiert, so dass sie sich bei der Umwandlung in das Programm kopieren lassen.

#### 3.2. Bildschirmformate (Maps)

Wir benutzen das IBM-Lizenzprogramm «Screen definition facility» (SDF) für das

- Entwickeln und Anpassen von Maps im Dialog, unabhängig von der Programmierung der Transaktionen
- Verwalten von Maps in besonderen Bibliotheken
- Überführen alter BMS-Maps in das neue SDF-Format

Eine derart entwickelte Map kann in beliebigen Transaktionen verwendet werden und vereinfacht die Kommunikation zwischen On-line-Programm und Datenstation.

#### 3.3. Zugriff auf Datenbanken

Alle Zugriffe auf DL1-Datenbanken gehen über den sogenannten DBIO-Vermittler als Interface. Es handelt sich dabei um ein von uns erstelltes Modul, das folgende Aufgaben übernimmt:

- Vornahme allgemeiner Kontrollen und Meldung von Unstimmigkeiten
- Sammeln der Daten aus der Datenbank entsprechend der verlangten logischen Informationseinheit, unter Verwendung normaler DL1-Funktionen
- Nachführen der Datenbank und der zugehörigen Beziehungen entsprechend der verlangten Funktion, das heisst Verändern, Abspeichern oder Löschen
- Übergabe von Informationseinheit und Retourcode an das Anwendungsprogramm

Aus unseren bisherigen Erfahrungen können wir die folgenden wesentlichen Vorteile nennen:

- Die klare Schnittstelle zwischen Anwendung und DB ergibt eine gewisse

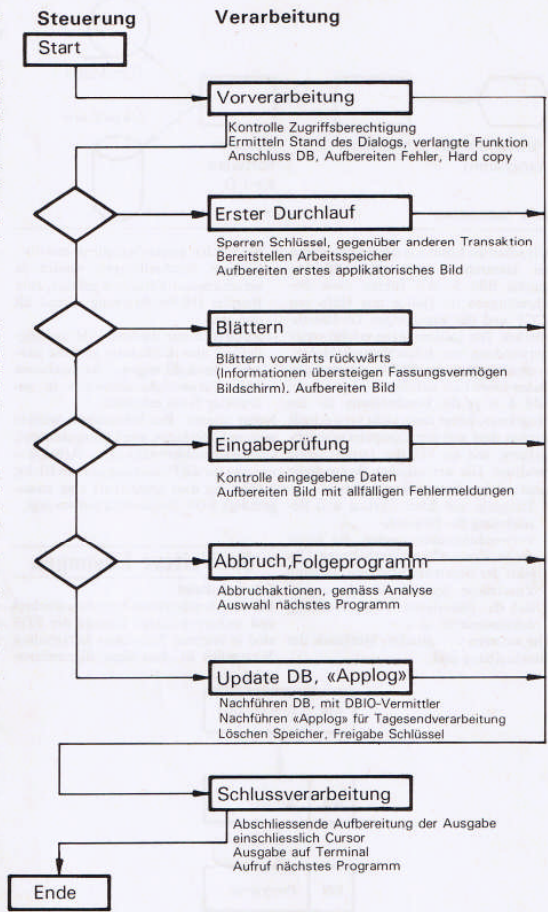


Bild 3 Normiertes On-line-Programm («Norop»)

Datenunabhängigkeit und Änderungsfreundlichkeit

- Wir haben eine bessere Gewähr für die Ordnungsmässigkeit der Datenbanken
- Der DBIO-Vermittler kann einmalig und unabhängig von den eigentlichen Programmen entwickelt und ausgetestet werden, was letztlich eine kürzere Einführung und höhere Produktivität ergibt

### 4. EDV-Dokumentation

Probleme bei der Anpassung von Programmen oder Dateien - in welcher EDV-Abteilung gibt es diese nicht - rufen nach einer Dokumentation, welche sowohl die spezifischen Belange eines Objektes oder Sachverhaltes darstellt, wie auch dessen Beziehungen im gesamten System zeigt.

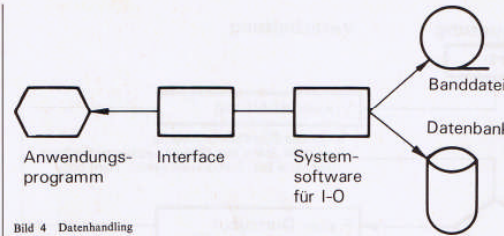


Bild 4 Datenhandling

Wir sehen die Lösung in der Beschreibung der hierarchisch geordneten Objekte gemäss Bild 5. Wir führen diese Beschreibungen im Dialog mit Hilfe von ICCF und der zugehörigen On-line-Bibliothek. Die Dateneingabe erfolgt unter Verwendung von Bildschirmshablonen, so dass die für jedes Objekt notwendigen Daten formatiert anfallen. Bild 6 zeigt die Beschreibung für ein Programm, wobei diese nicht verwechselt werden darf mit der Computerproblemstellung, wie sie aus der Detailanalyse resultiert. Die wesentlichen Bestandteile einer solchen Beschreibung sind:

- Titelzeile, mit Identifikation und Bezeichnung des Objektes
- Verwendungskomponenten, das heisst die in einem Objekt enthaltenen Objekte der hierarchisch tieferen Ordnung
- Zusätzliche Spezifikationen, so dass sich das betreffende Objekt eindeutig dokumentieren lässt

Die weiteren wesentlichen Merkmale der Beschreibung sind:

- Sie werden projektbegleitend erstellt
- Gewisse Beschreibungen werden in verschiedenen Versionen geführt, zum Beispiel DB-Beschreibung: Stand alt und neu
- Zusätzlich zur direkten Abfragemöglichkeit über Bildschirm sind die ganzen Beschreibungen, Verzeichnisse oder Verwendungsnachweise in gedruckter Form erhältlich

Nebst diesen Beschreibungen werden weitere Unterlagen, wie Lösungskonzept, Computerproblemstellung, Arbeitsanweisung für die Produktion usw. im Dialog geführt, so dass gesamthaft eine aussagefähige EDV-Dokumentation vorliegt.

### 5. Weitere Lösungen

5.1. Richtlinien  
Die hier beschriebene Verfahrenstechnik und weitere wichtige Belange der EDV sind in internen Richtlinien festgehalten. Wesentlich ist, dass diese allgemeiner-

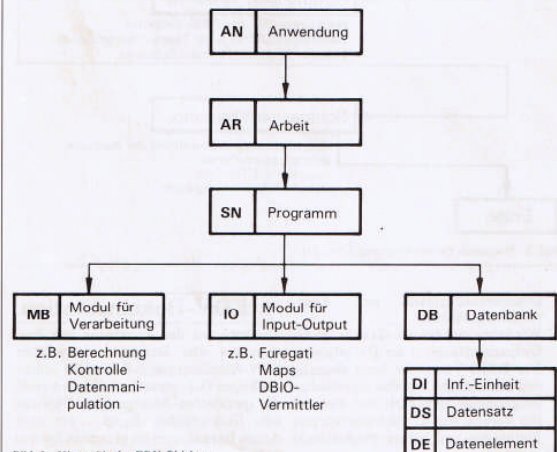


Bild 5 Hierarchie der EDV-Objekte

ständig gehalten und laufend nachgeführt sind, so dass sie als Arbeitsinstrument verwendbar sind.

#### 5.2. Aktenarme Sachbearbeitung

Auch in der EDV wird nach Möglichkeit die aktenarme Sachbearbeitung angestrebt. Wir benutzen das IBM-Lizenzprogramm «Interactive computing and control facility» (ICCF) mit eigens erstellten Bildschirmshablonen zur Benützerführung für die

- On-line-Programmierung und Dokumentation, wie in den vorangehenden Kapiteln beschrieben
- Vorbereitung und Steuerung der Bibliotheksverwaltung, so dass unser Programmmaterial über mehrere Stufen sichergestellt und die Originalversion gegen unbefugten Zugriff geschützt ist
- On-line-Avor und Übergabe von Arbeiten zur Stapelverarbeitung unter «Power»

#### 5.3. Informations- und Kontrolldaten

Wir führen eine besondere Datenbank mit allgemeingültigen Informationen, wie Länder, Währungen, Tarife. Die geringere Mutationshäufigkeit im Informations- und Kontrollbereich «Ikon» hat uns bewogen, eine parametrisierte Programmierung zu wählen, welche erlaubt, neue Tabellen mit geringem Aufwand zu integrieren, dafür aber weniger Komfort bei der Dateneingabe bietet. Der «Ikon» lässt sich tagsüber im Dialog mutieren. Die Mutationen werden allerdings auf einem Arbeitsbereich vorgenommen, der abends in die produktiven Daten eingespielt wird. Dadurch erhalten wir eine klare Abgrenzung für die Inkraftsetzung neuer Basisinformationen und eine gewisse Unabhängigkeit zwischen Programm und Daten.

#### 5.4. Applikatorischer Log

Die erledigten Datenbankmutationen sowie Transaktionen mit einer Folgebearbeitung werden im applikatorischen Log «Applog» eingetragen. Dieser dient folgenden Zwecken:

- Steuerung der Tagesendverarbeitung, wobei die applikatorisch notwendigen Daten aus den entsprechenden Datenbanken entnommen werden
- Nachweis der Datenbankveränderungen im zeitlichen Ablauf, zur Klärung allfälliger Unstimmigkeiten oder für die Statistik, je nach Bedarf

#### 5.5. Zugriffsberechtigung

Wir beschränken uns vorerst auf die in der Software vorhandenen Einrichtungen, und zwar

- Anmeldeverfahren, Sicherheitscode und Benützerklasse im CICS, Programmprodukt für die computergetriebene Sachbearbeitung der Benützer

"SCHWEIZ" \* \* BZSN / 2  
 ALLGEMEINE \* BESCHREIBUNG DER PROGRAMME \* 25.09.81 / V00  
 VERSICHERUNGEN \* "NOVITA" \* J. VESELY

BEZ: MUTATION ADRESSDATEN SPRACHE: COBOL IDENT: SN207

KURZBESCHREIBUNG:

ONLINE MUTIEREN ALLER BESTEHENDEN ADRESS-DATEN IN DER  
 ADRESS-DB MITTELS MUTATIONS-CODE.  
 AUSGENOMMEN SIND GESCHAFTSVERBINDUNGS-MUTATIONEN.  
 ZUGEHÖRIGE SEKUNDAER-INDICES WERDEN DURCH DBIO-  
 VERMITTLER ANGEPAST.

VERSION: 00 UEBERN-DATUM: 31/07/81 ANALYST: KRH PROGRAMMIERER: HH

VERW.KOMP.	BEZEICHNUNG	TECHNISCHE SPEZIFIKATIONEN	
**** DATENBANKEN		ZUGRIFF	DATENTRAEGER
X.DB959	ADRESS-DB	UPDATE DIREKT PLATTE	
X.DB659	APPLIKATORISCHER-LOG	UPDATE DIREKT PLATTE	
X.DB966	SEKUNDAER-INDEX-DB	UPDATE DIREKT PLATTE	
**** DBIO-VERMITTLER			
X.MV102	DBIO-VM. ADRDB ONLINE		
X.MV101	DBIO-VM. APPLLOG ONLINE		
**** INFO-EINHEITEN			
X.DI301	ADRDB-INFO		
X.DI401	APPLLOG-INFO		
X.DI501	SEKID-INFO		
**** MAPS			
X.MM207	MAP ADRESS-MUTATION		

Bild 6 Programmbeschreibung

- Anmeldung mit Passwort, Zuteilung der Bibliotheken und Kennzeichnung der Bücher als privat oder öffentlich im ICCF, Programmprodukt für die Entwicklungstätigkeit in der EDV

Dadurch lassen sich in angemessener Weise Fehlmanipulationen vermeiden sowie Zuständigkeits- und Verantwortungsbereiche sauber abgrenzen. Die Lösung stellt unseres Erachtens einen vertretbaren Kompromiss dar zwischen Aufwand und Sicherheit.

**5.6. Wiederherstellung und Wiederanlauf des Systems**

Während der Verarbeitung werden die wesentlichen Systemereignisse wie  
 - Veränderungen an Datenbanken  
 - Start und Ende von Programmen  
 durch das CICS in einem technischen Log festgehalten. Dieser technische Log, eine frühere Sicherstellung der DB sowie die entsprechenden IBM-Dienstprogramme ermöglichen Recovery und Restart. Allerdings überwachen wir aufgrund von Steuer- und Kontrollinformationen im Record Null der Datenbank mit Hilfe selbstentwickelter programmierter Kontrollen:

- Den ordnungsgemässen Stand und die übereinstimmende Version zwischen Datenbank und Programm
- Die Vollständigkeit und richtige Sequenz der technischen Logs

**6. Zusammenfassung**

Entscheidend ist, von Anfang an eine saubere technische Infrastruktur für die DB/DC-Verarbeitung zu erstellen, und zwar vorerst zur einwandfreien Lösung der täglich anfallenden Probleme.

Wie dieser Beitrag zeigt, ist dies möglich unter Einsatz standardmässiger IBM-Software in Verbindung mit eigenen organisatorischen Massnahmen und vertretbarem Aufwand. Daraus resultiert eine gute Effizienz und Transparenz in den verschiedenen Bereichen der EDV.

Wir beabsichtigen, vorerst mit dieser Lösung weitere praxisbezogene Erkenntnisse zu sammeln. Die Weiterentwicklung zu einem späteren Zeitpunkt erscheint uns aber durchaus möglich.

**Efficiency et transparence de l'informatique**

L'informatique a normalement pour but de permettre à l'utilisateur de rationaliser ou tenir à disposition les informations utiles à la conduite de l'entreprise. Pour satisfaire les exigences des années 80 une infrastructure technique claire est indispensable même dans le domaine de l'informatique. Ceci est réalisable à des frais convenables et rapidement compensés par une efficacité et une transparence accrues.

# Endlos-Formular-Kriterien

- 1. Full-Service**
- 2. Allround-Durchschreibe-Technik**
- 3. Black-outs nach Wunsch**
- 4. Back-Prints**
- 5. Separability**

Verlangen Sie Auskünfte über die Vorteile und Kombinations-Möglichkeiten dieser Techniken und Offerten von den Spezialisten der

**wagner ag**

An die Firma Wagner AG,  
 Badenerstrasse 820,  
 8048 Zürich (Tel. 01/62 16 00)

Wir interessieren uns für  
 weitere Auskünfte zu 1., 2., 3., 4., und 5.  
 Offerte für Neudruck nach beiliegendem Layout/Muster.  
 Auflage \_\_\_\_\_

Firma: \_\_\_\_\_  
 Adresse: \_\_\_\_\_  
 Name: \_\_\_\_\_