



The Development and Deployment of a Workflow System partially written in Ada95

Frank Piron, frank.piron@konad.de

KonAd GmbH, In der Reis 5, D-79232 March-Buchheim



Content

- Topic and Terms
- Why Ada95?
- Library Development
- Application Development
- Application Structure
- Experiences
- Conclusion

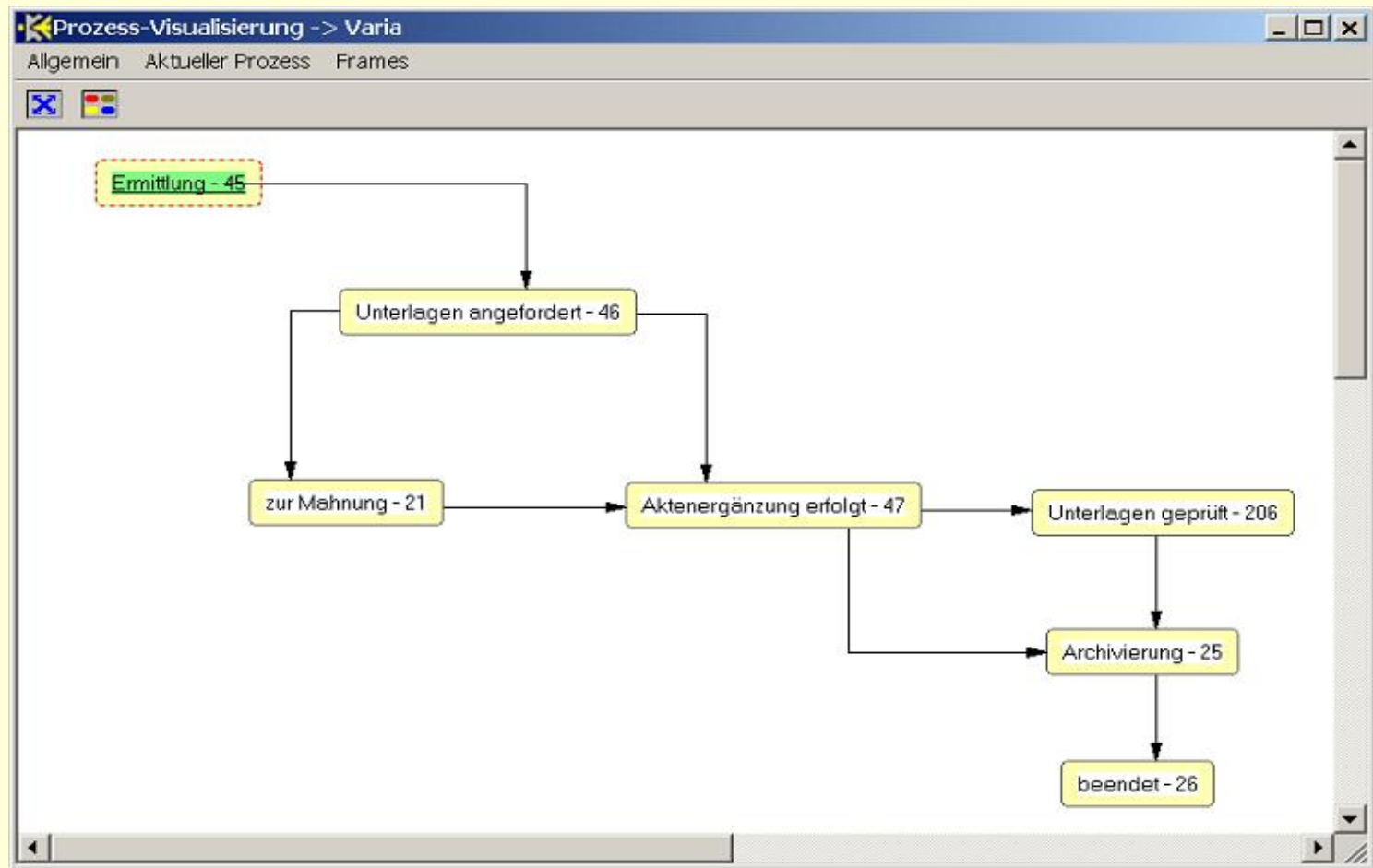
Topic and Terms

Development of a State of the Art Workflow System including

- ✓ Flow Control with Process Design and Rule Engine
- ✓ Document/Content – Management
- ✓ Rich win32 Client for Users and Administrators
- ✓ Interfaces to Electronical Archives, MS-Office and other Applications

Topic and Terms

Flow Control and Process Design



Business Cases are controlled by Processes

Topic and Terms

Document/Content – Management

The screenshot illustrates a document management system interface. The main window, titled 'Elektronischer Schreibtisch', shows a list of documents with the following data:

Typ	Geschäft	Verzeigter	Anmerkungen	Aktiv.	GK	Datum	Dok.
	ANF-0049		test2	Ausk_b		10.03.2006 10:09	1
	ANF-0050		test	Ausk_b		13.03.2006 09:58	2

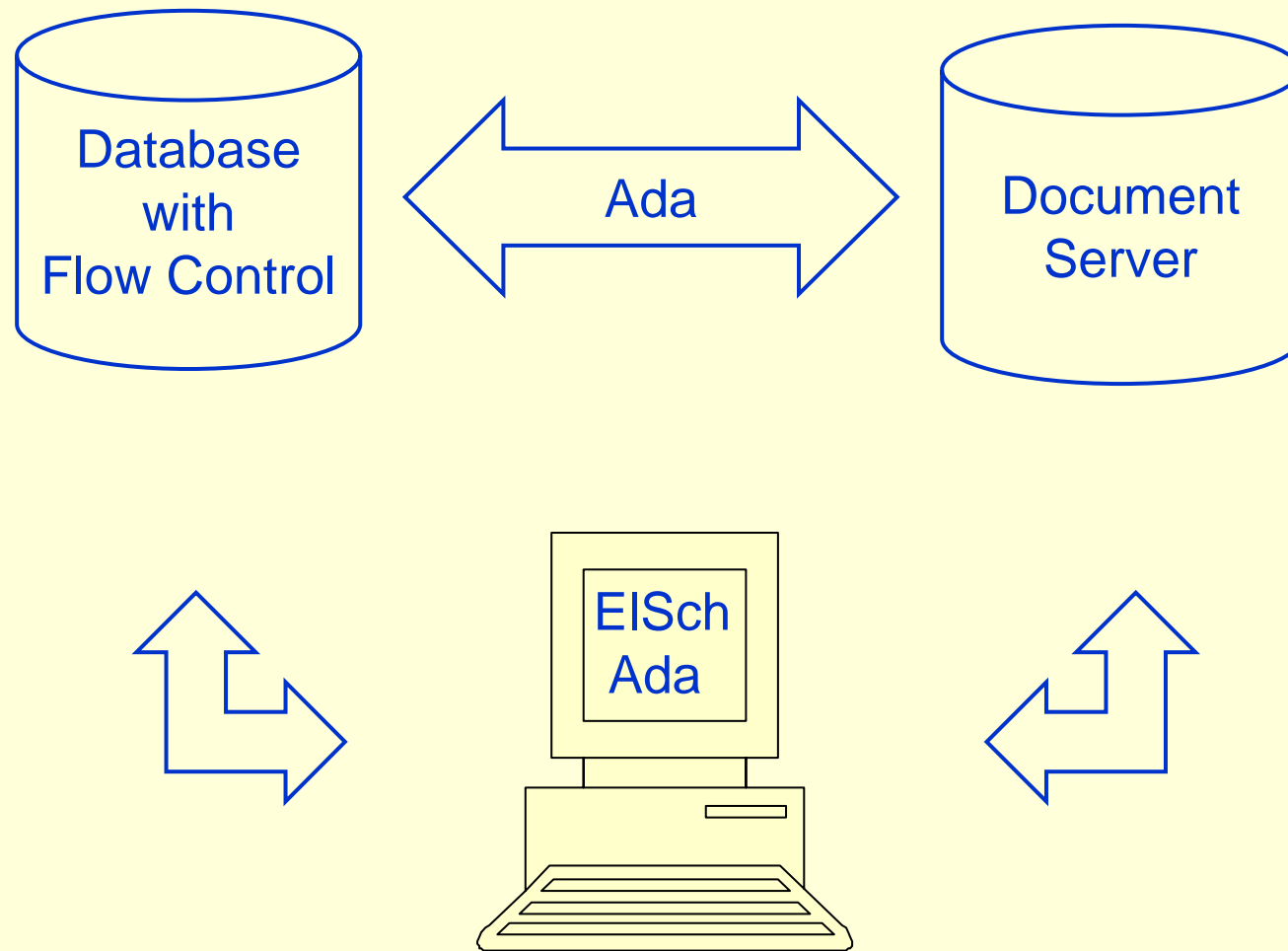
Below this, a 'Dokumentenliste für Geschäft ANF-' window displays a table with the following data:

Typ	Einlaufzahl	Angelegt von	Dok.-Typ
	65	GmbH, KonAd	Import
	23	GmbH, KonAd	Aktenver

The Microsoft Word window shows a document with the text 'Peter Muster' and 'Freiburg, 21. Nov'. At the bottom of the Word window, the text 'Sehr geehrte Frau Schmitz,' is visible.

Every Business Case has a set of attached Documents

Architecture



Why Ada95?

KonAd Programming Background in 2001

- Oracle PL/SQL Server programming
- Oracle Client-Server Tools Development
- C, Visual Basic as General Purpose and „Glue Code“ Languages

Why Ada95?

Need for Change/Extension because

- Use of System Services difficult
- Oracle cancelled further development of PL/SQL based Client Tools by end of the nineties and switched to JAVA
- Homogeneous programming environment

Why Ada95?



- PL/SQL and Ada are similar, since PL/SQL (1991) was derived from Ada83
- Ada95 is suitable to build big reliable Software Systems
- Ada95 is an international Standard

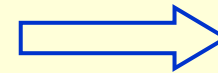
Library Development

2002 - ...

2002

Learning Ada95 and decision to use GNAT and Emacs

We needed Libraries for Database-Interaction and for Win32 GUI Development

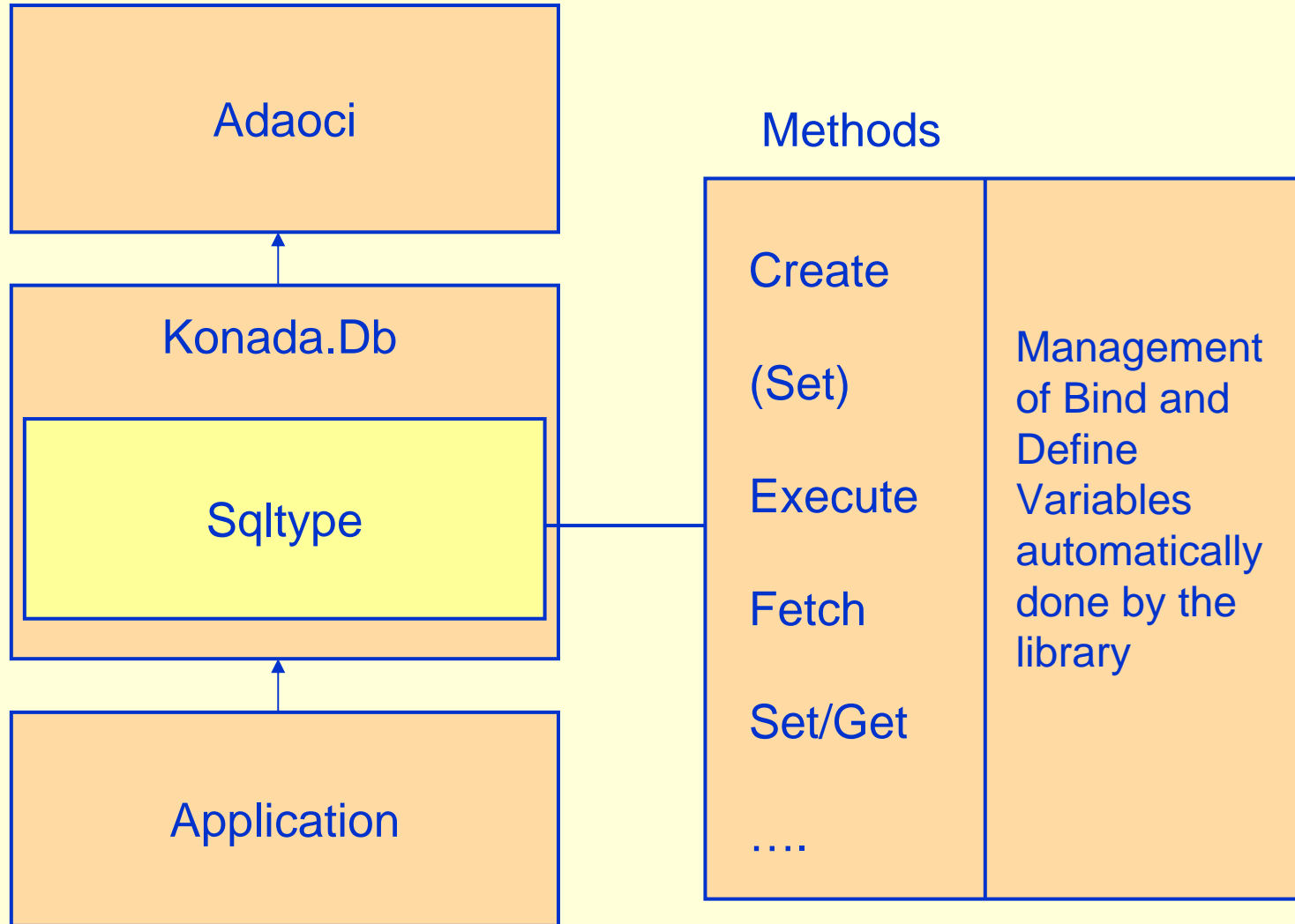


Web Research and Evaluation

- ✓ **Adaoci** (Dmitriy Anisimkov) as a starting point for an Oracle Database Access Layer
- ✓ **GWindows** (David Botton) as a Basis for GUI Development

Library Development

Konada.Db based on Adaoci



Library Development

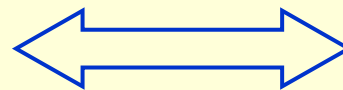
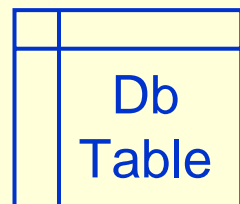
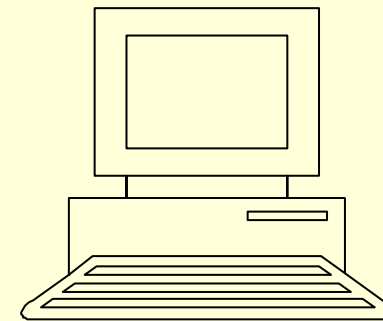
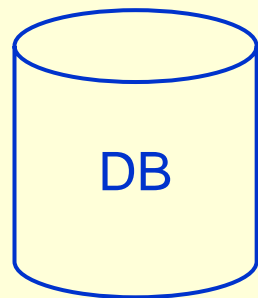
Konada.Db - First Step: Sqltype

```
emacs@FRANK
File Edit Options Buffers Tools Ada Help

declare
  Person: Sqltype;
begin
  -- Konada.Db.Sql
  Create(Person,
    "select * from emp where empno=:empno_to_find");
  -- Statement is prepared
  -- Bind variables are known now
  -- set :empno_to_find in sqltype-instance Person
  -- bind is implicit here
  Set(Sqlcmd => Person,
    Position => 1,
    Value => 7369);
  Execute(Person);
  Fetch(Person)
    declare
      Name: String:=Get(Person,"ename");
    begin
      Text_Io.Put_Line(Name);
    end;
end;
```

Library Development

Konada.Db – Next Steps: Rowtype and Tabletype



Read and modify
without SQL

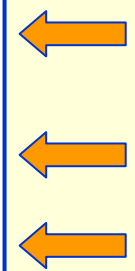
Library Development

Konada.Db - Next Steps: Tabletype

```
tabletest.adb
File Edit Options Buffers Tools Ada Help

declare
  Employees: Tabletype;
  Rows_Fetched: Natural:=0;
begin
  -- logon to the database
  Logon("scott/tiger@sun");
  -- create clientside table object
  -- for database table "emp"
  Create_From_Db_Table(Table => Employees,
                      Db_Table_Name => "emp");

  -- allow updates
  Grant_Access(Employees, Update);
  -- fill table with all rows
  Fetch_All(Employees, Rows_Fetched);
  -- set salary of the first two emps to 1500 $
  Set(Table => Employees, Name => "sal", Value => 1500.0);
  -- move to the next row
  Move(Table => Employees, To => Next);
  Set(Table => Employees, Name => "sal", Value => 1500.0);
  -- post changes to the database
  Post_Changes;
  -- and commit
  Commit;
end;
```



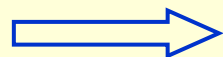
Library Development

GWindows_Extended

What is GWindows?

GUI centered Ada95 Binding to the Win32-API written by David Botton

- ✓ Easy to use
- ✓ Clearly written
- ✓ Easy to extend
- ✓ Free (GMGPL)

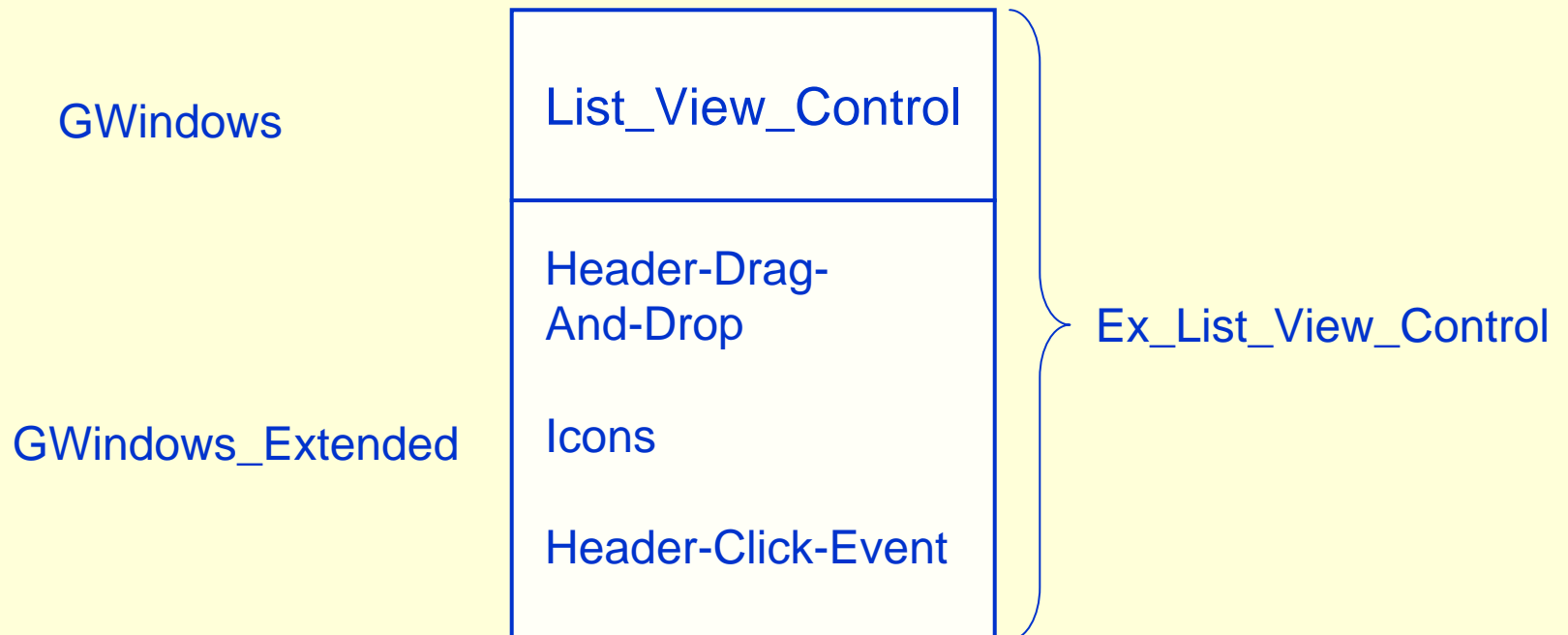


2003: Decision to use GWindows

Library Development

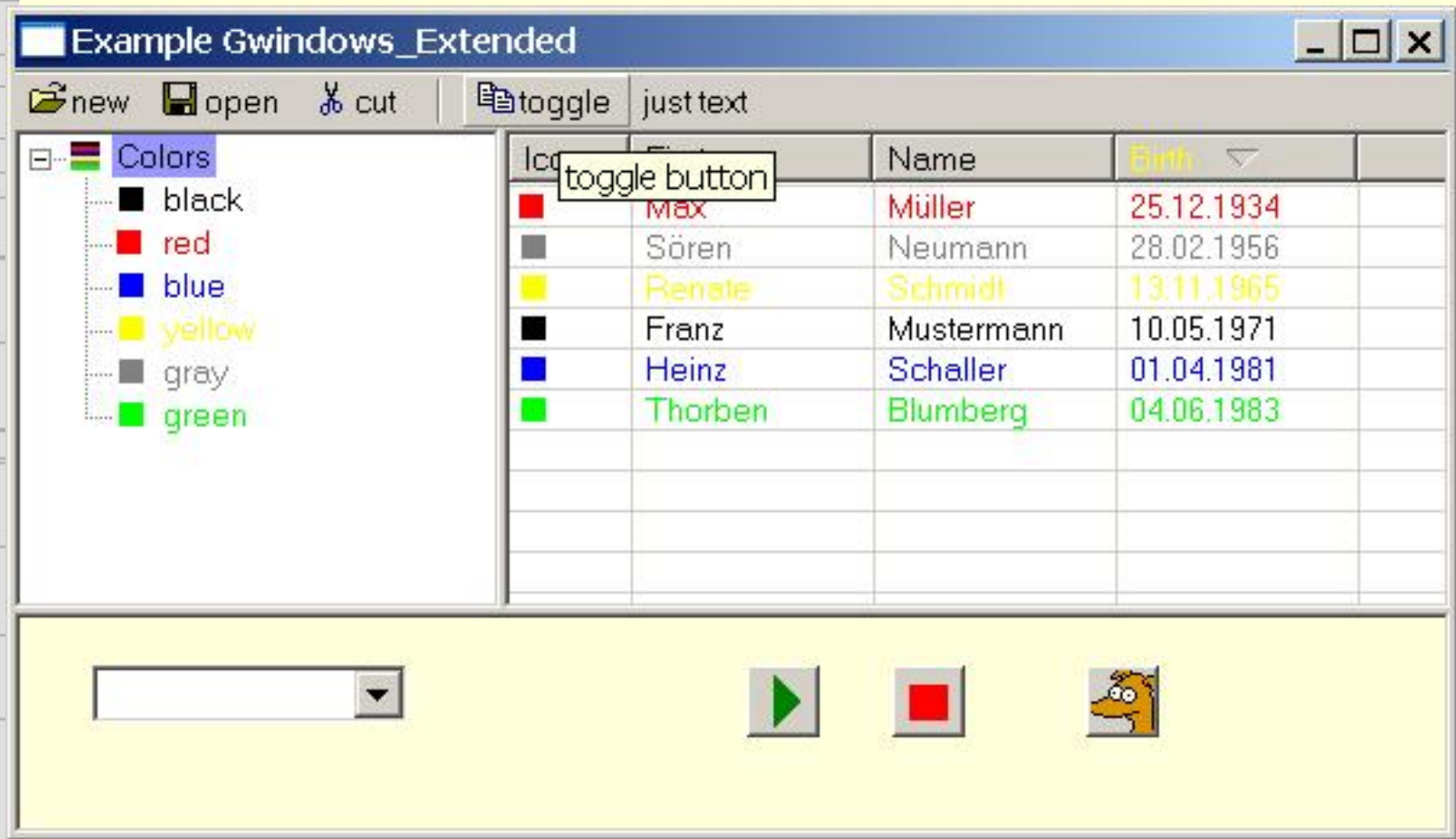
GWindows_Extended

- New Types of Controls derived from GWindows Types with more features
- Some bug fixing
- Some Procedures and Utilities added



Library Development

GWindows_Extended

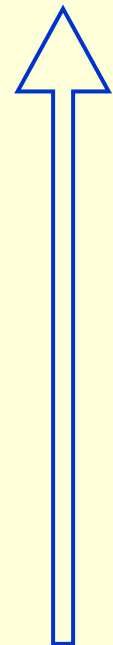
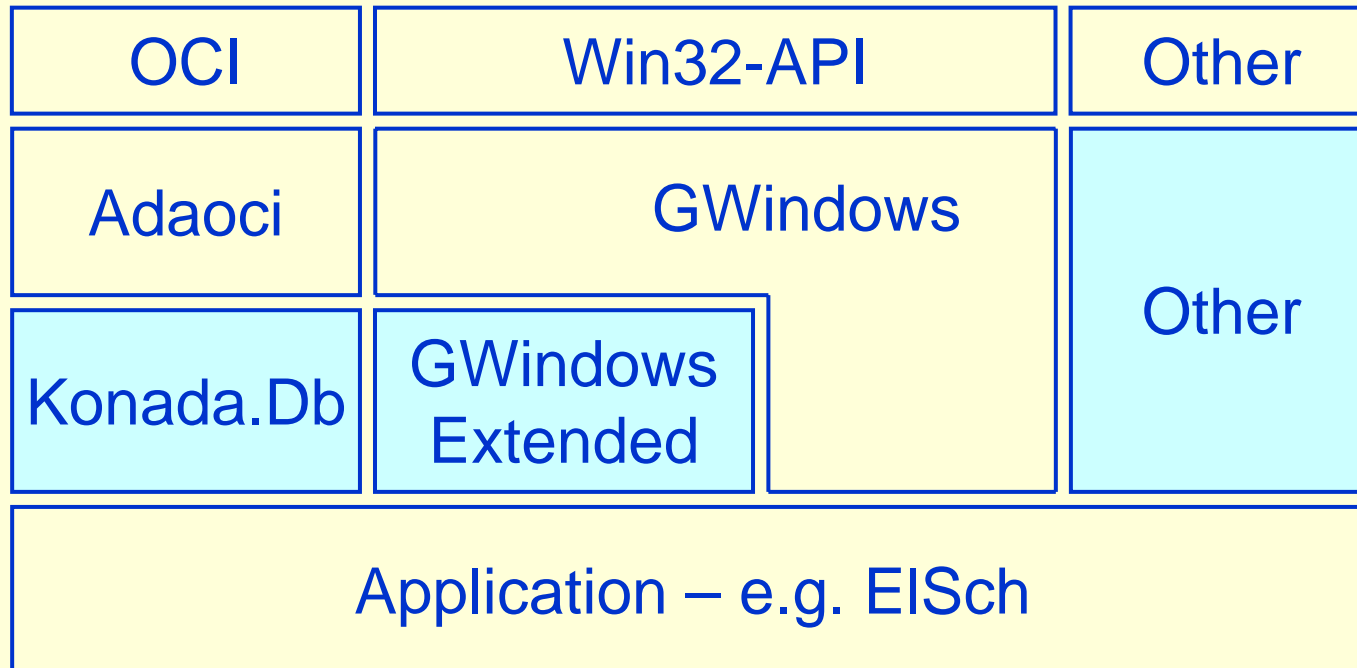
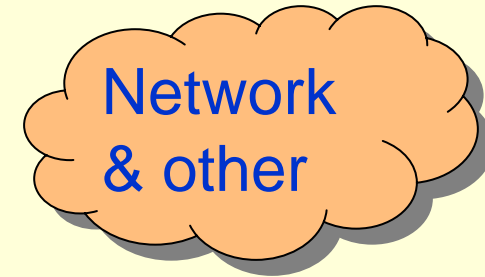


Library Development

GWindows_Extended

```
emacs@FRANK
File Edit Options Buffers Tools Ada Help
declare
  Combo: Ex_Drop_Down_Combo_Box_Type;
begin
  -- combo box
  Create(Combo => Combo, Parent => Win_bottom,
        Text => "",
        Top => 20, Left => 30,
        Width => 130, Height => 200,
        Sort => false);
  Set_Selection_Field_Height (Combo, 18);
  declare
    Index: Integer;
  begin
    Add(Combo, "January", Index);
    Add(Combo, "February", Index);
    Add(Combo, "March", Index);
    Add(Combo, "April", Index);
    Add(Combo, "May", Index);
    Add(Combo, "June", Index);
    Add(Combo, "July", Index);
    Add(Combo, "August", Index);
    Add(Combo, "September", Index);
    Add(Combo, "October", Index);
    Add(Combo, "November", Index);
    Add(Combo, "December", Index);
  end;
  Set_Autocomplete(Combo, True);
  Force_Invalid_Text(Combo, True);
end;
-1\** example_pkg.adb (Ada CUS-1.3)--L266--77%-----
```

Library Development



Application Development

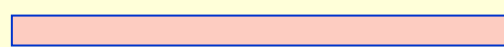
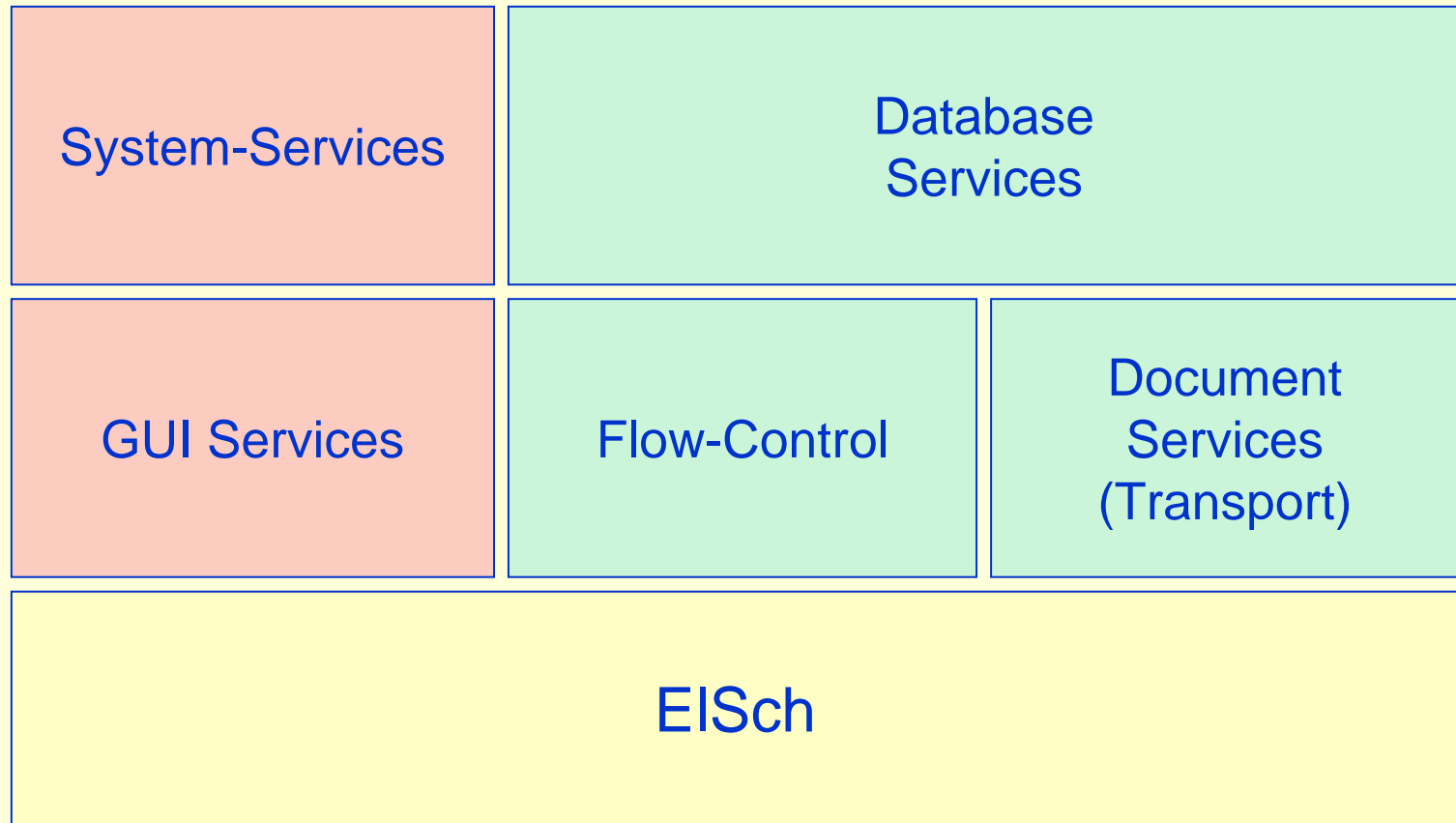
2004,
2005

Primary Goal

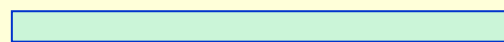
Separation of functional Areas

- EISch ↔ User Interface
- GUI-Functions and Services
- Flow-Control Functions
- Document Functions
- Utilities and System Services

Application Structure



Depends on Platform



Platform independent,
(used on Server also)

Application Structure

Example: Load a Baskets Content

The screenshot shows the 'Elektronischer Schreibtisch' application window. The title bar reads 'Elektronischer Schreibtisch'. The menu bar includes 'EISch', 'Arbeitskorb', 'Geschäft', 'Dokument', 'Programme', and 'Hilfe'. The toolbar contains various icons for navigation and actions. The left pane shows a folder tree for 'KonAd GmbH' with sub-items: 'Persönlicher Ordner', 'Arbeitsfläche', 'Eingangskorb' (highlighted), and 'Suche'. An arrow labeled 'Klick' points to the 'Eingangskorb' folder. The right pane displays a table with the following data:

Typ	Geschäft	Verzeigter	Anmerkungen
	2005-000-128	Piron Franz	
	2005-000-118	Piron Franz	
	2005-000-116	Piron Franz	Basket
	2005-000-115	Piron Franz	
	2005-000-109	Maier Falk	
	2005-000-108	Piron Franz	
	2005-000-106	Maier Falk	

At the bottom of the window, a status bar shows 'Eingangskorb', '601', 'Geladen', and 'sun'.

Application Structure

Example Load-Basket: Flow of Control

Procedure/Function

Package

„User clicks on a Node“

Do_On_Click(Tree.Node)

Load_Basket(This_Basket)

Bc_Data(Data, Bc.Id)



Add_To_List(Data, Basket.List)

Add_Item(...)



EISch.Callbacks

EISch.Basket

Wfl.Db

EISch.Basket

„GWindows_Extended“



Application Structure

Application Size

- Serverside Code (PL/SQL)
60_000 Lines
- Serverside Code Ada95
10_000 Lines
- Clientside Code Ada95
80_000 Lines

Experiences

Oracle PL/SQL → Ada95

- Where is my good old varchar2 Datatype?
- Programming on the Stack
- From Oracle Namespaces to Ada95 Visibility
- Emacs and Gnat vs. Oracle Tools
- Do we really need a GUI-Builder?

Experiences

Oracle PL/SQL → Ada95

```
create or replace
procedure Varchar2_Demo
is
  Message varchar2(20) := '';
begin
  Message := 'Hello World!';
  Dbms_Output.Put_Line(Message);
end;
```

Experiences

Learning Ada95

- John Barnes: Programming in Ada95
- Burns/Wellings: Concurrency in Ada
- „Ada distilled“ by Richard Riehle
- C.L.A. Newsgroup, very helpful!
- Difficult: Visibility and Dispatching
- Easy to grasp: Ada95 Tasking
- We learned: Ada ist a Language full of Thought.

Experiences

Interfacing with C

- Interfacing with the win32 API
Sometimes Representation Clauses necessary.
- Interfacing with Oracle OCI
easy since OCI is designed platform independent
- UNIX System Calls
easy

Conclusion

- Hard Work
Learning and building Libraries
- Teaching Overhead
- Efficient Development Now possible
- The System runs in its first Production Environment since 9/2005 with no major Errors



Recent/FutureDevelopment

- Dynamically layouted GUI-Objects for Database Interaction
- Rule-Engine API
- AWS-Development

Recent/Future Development

Übergangsregeln

Name: Lageort alt:

Funktionsname: Lageort neu:

Kategorie: Aktivitaet alt:

Aktiviert: Aktivitaet neu:

Modus: Erzeugt am:

Prioritaet: Erzeugt von:

Beschreibung: Zuletzt geaendert:

Prozess alt: Geaendert von:

Prozess neu:

NAME	FUNKTIONSNAMEN	KATEGORIE	AKTIVIERT	MODUS	PRIORITAET	BESCH
tesrgele neu	TEST_RULE	Geschäftsfall weiterleiten	Ja	Vorher	1	
testregel	TRANSFER_BEARBEITEN_RULE	Geschäftsfall weiterleiten	Ja	Vorher	1	
mhtfjghf	TRANSFER_BEARBEITEN_RULE	Geschäftsfall weiterleiten	Ja	Nachher	1	

Thank You

