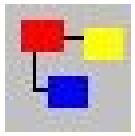
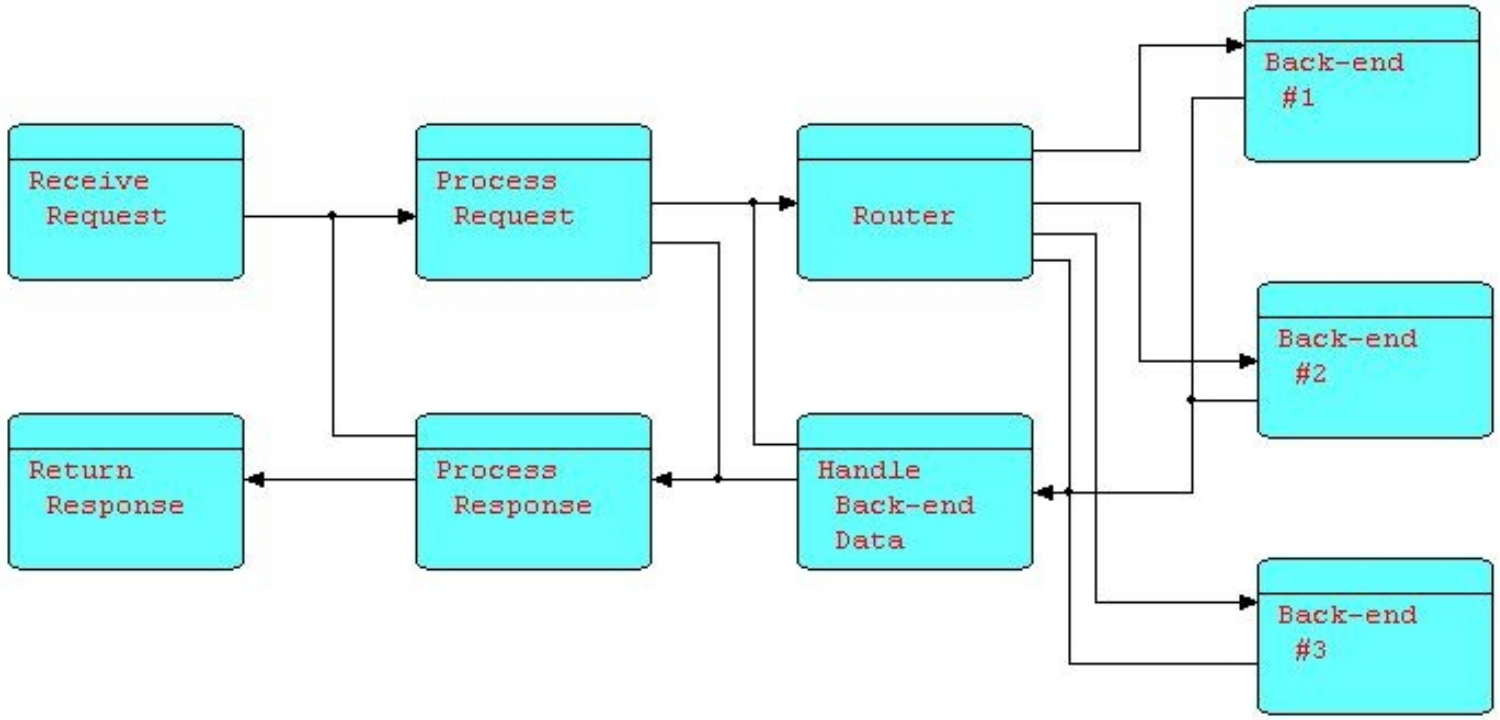


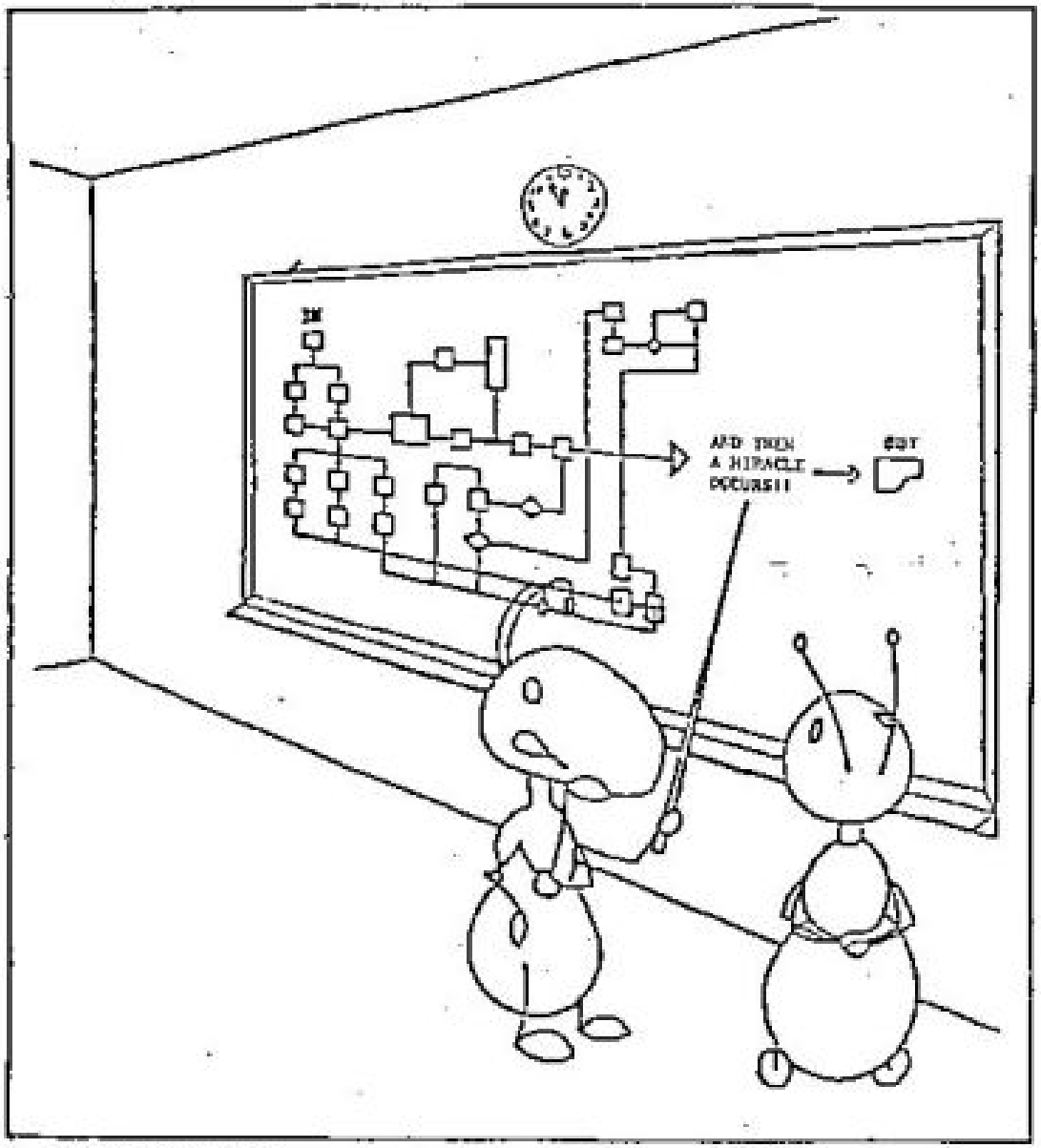
FLOW-BASED PROGRAMMING



2007 (!)

J.Paul Morrison

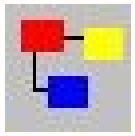
J.P. Morrison Enterprises Ltd.



GOOD WORK....
BUT I THINK WE NEED A LITTLE MORE DETAIL RIGHT HERE.

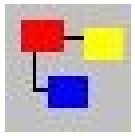
https://ibm.biz/BdZWuk

https://ibm.biz/BdZWuk



Symptoms of the Problem

- **Missed deadlines; budget overruns**
- **"Hidden" application backlog**
- **Fads and "snake-oil salesmen"**
- **Endless number of “new”, incompatible, tools and languages to be learned**
- **"Reinventing the wheel"**
- **Hard-to-maintain systems**

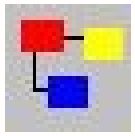


The *Real* Cause of the Problem: The von Neumann Model

- **Inappropriate for application design:**
  - **Code is procedural, sequential**
  - **One-step-at-a-time view of processing**
  - **Hierarchic structure of code**
  - **Subroutine call as building block**
- **Uniform array of read/write memory cells**
  - **Programmer has to control exact sequence of events**

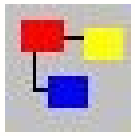
- **Procedural approach is rare in real life... and always difficult**

- **Real life is asynchronous, cooperative**



Maintainable Systems

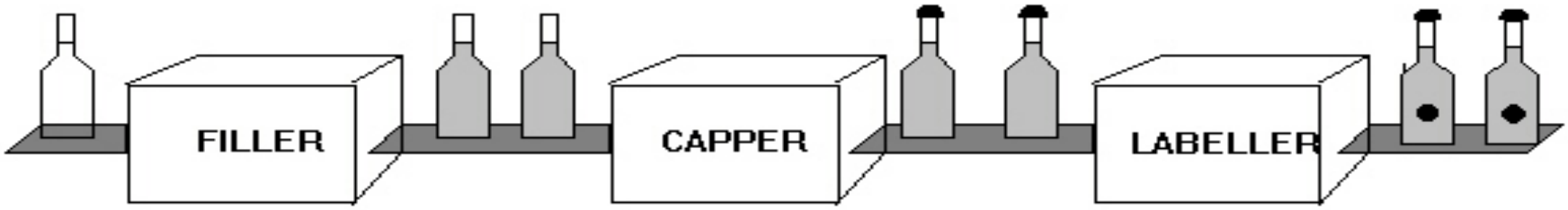
- **Express problem in terms of transforms on streams of data**
- **Most design technologies describe applications this way – then design has to be converted to procedural code**
- **"Unit Record" had these characteristics**
- **Precoded, pretested black-box ("LEGO") components have long been IT goal**



FBP Characteristics

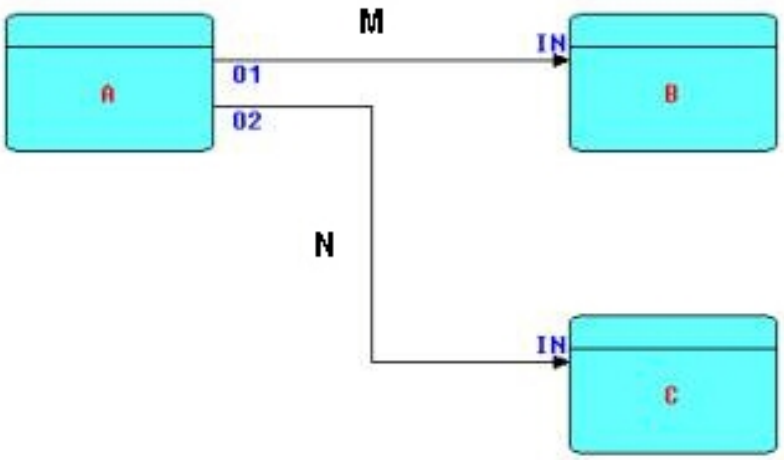
- **Asynchronous processes communicating via streams of data packets**
- **Data packets with a lifetime of their own**
- **Definition of connections *external* to components**
- **Consistent view from macro to micro, and from design to maintenance**

Soft Drink Bottling Factory



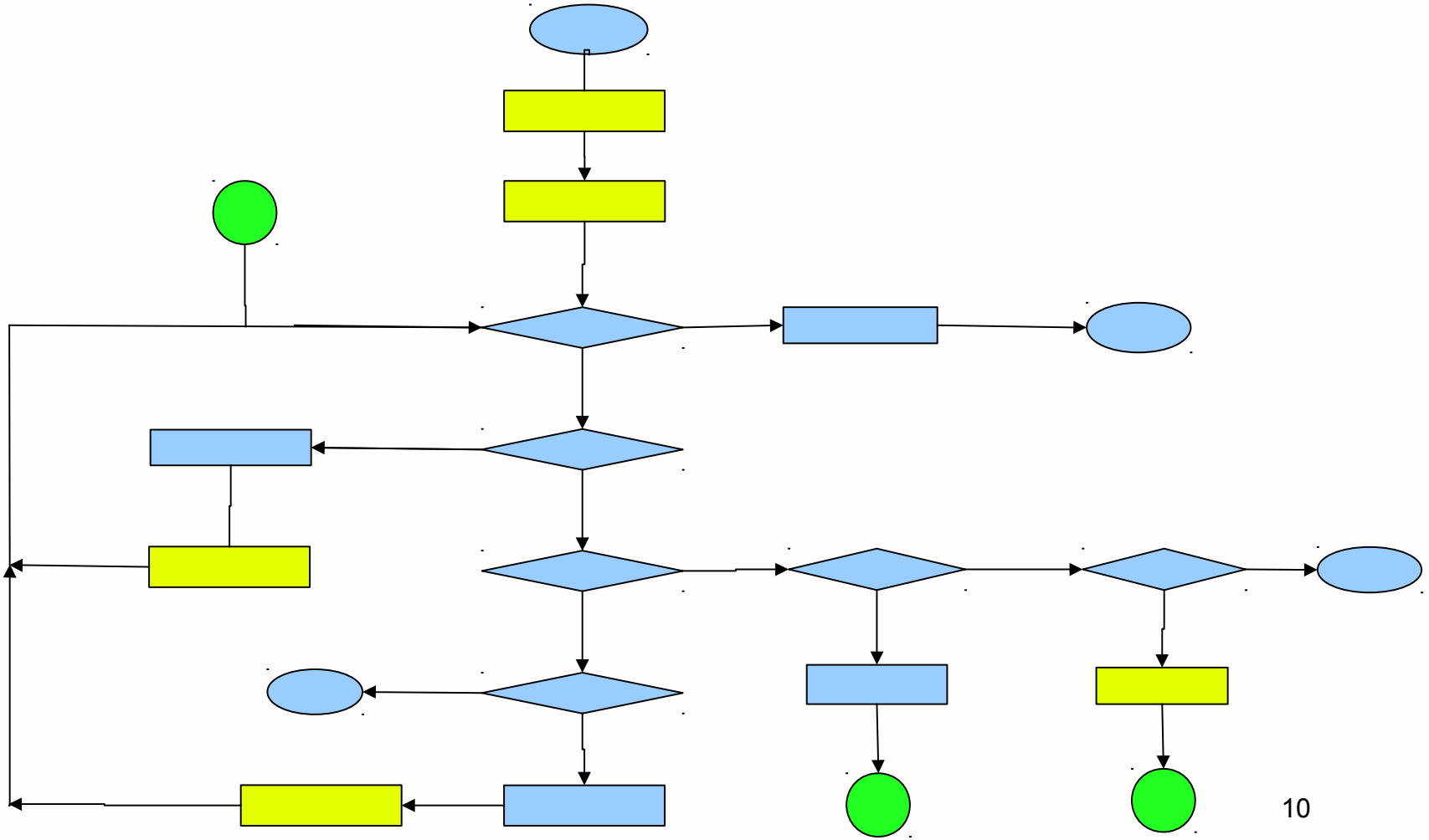
- Independent well-defined functions
- Clean interfaces
- Simple to reconfigure
- Minimizes side-effects
- Designer can sit at one “station”, or can follow an item through system

Elements of FBP

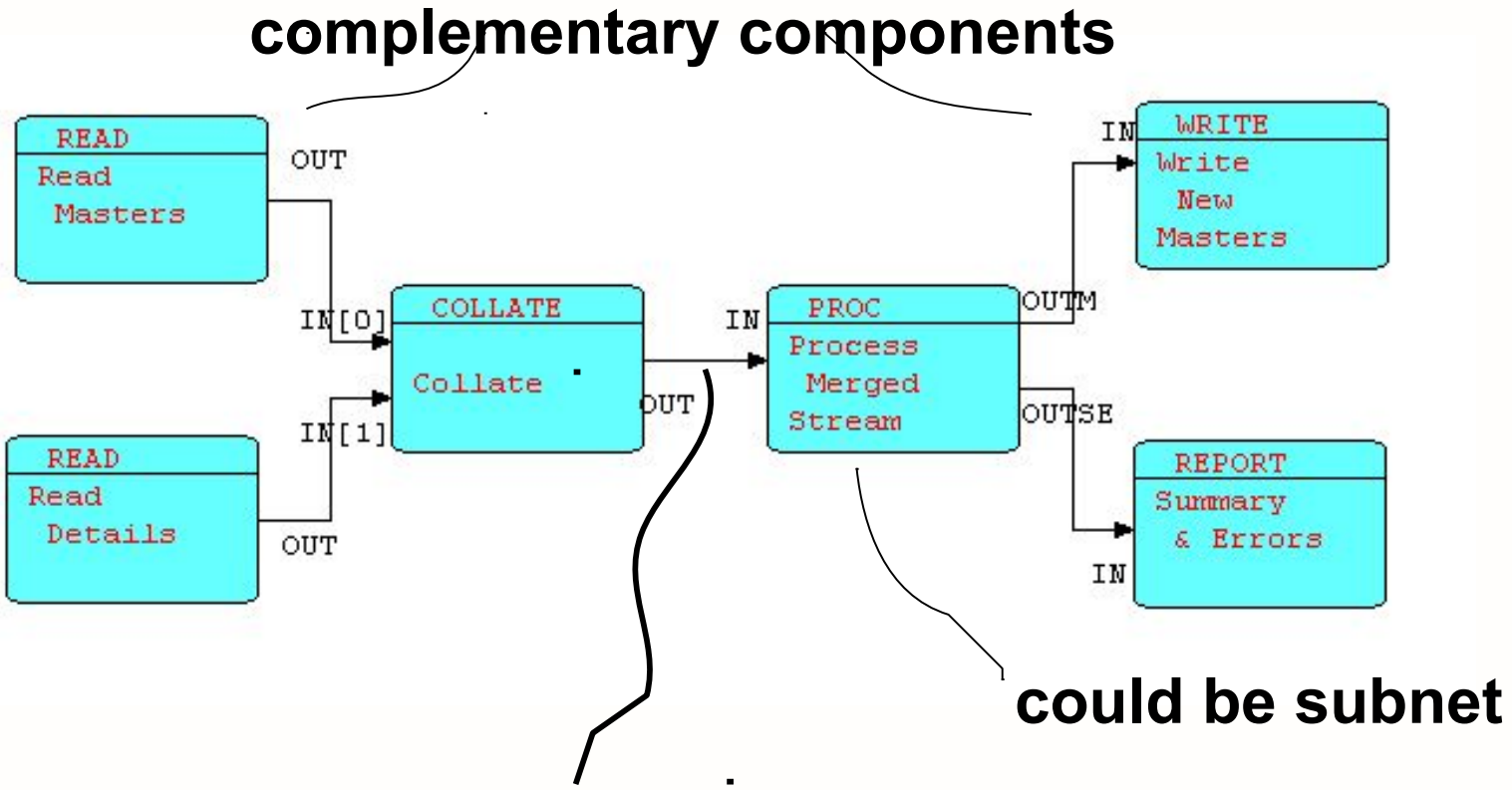


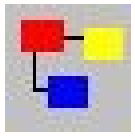
- Processes (A, B, C)
- Connections (M, N) - "bounded buffers"
- Ports (O1, O2, IN of B, IN of C)
- Connections defined externally to processes
- Streams of data chunks (Information Packets)

Traditional “Update” Application



FBP “Update” Application

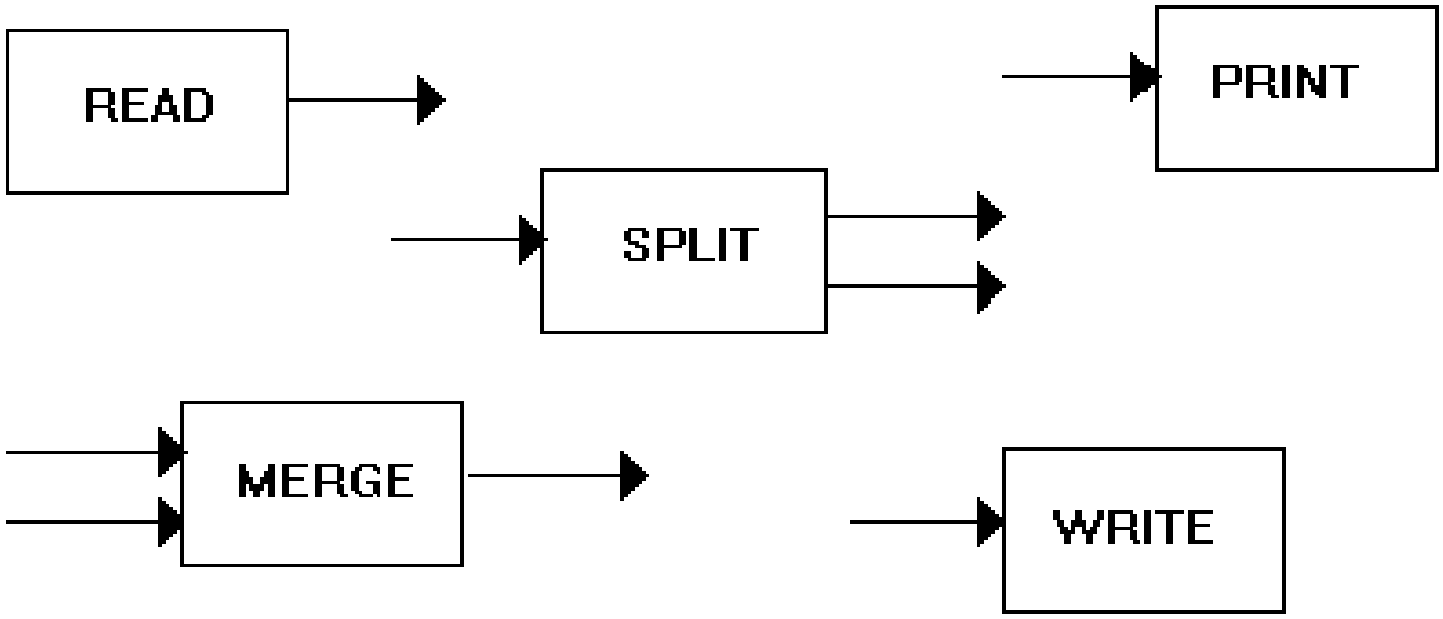




“Coordination Language” (*)

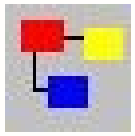
- Flow-Based Programming is a “coordination language”, not a programming language
- FBP is language-independent
- One language doesn't have to do everything
- (*) Term borrowed from “Linda” (Gelernter & Carriero - Yale)
- FBP is visual and component-oriented
- “Utilities” can be turned into components

“Configurable Modularity”



“Characteristic of any engineered system”

(Nate Edwards, IBM)

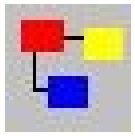


Component-Oriented

- **“Black boxes”**
- **Parametrization using “IIPs”**
- **Ports**
- **Parameters may be mini-language**

Compatible with:

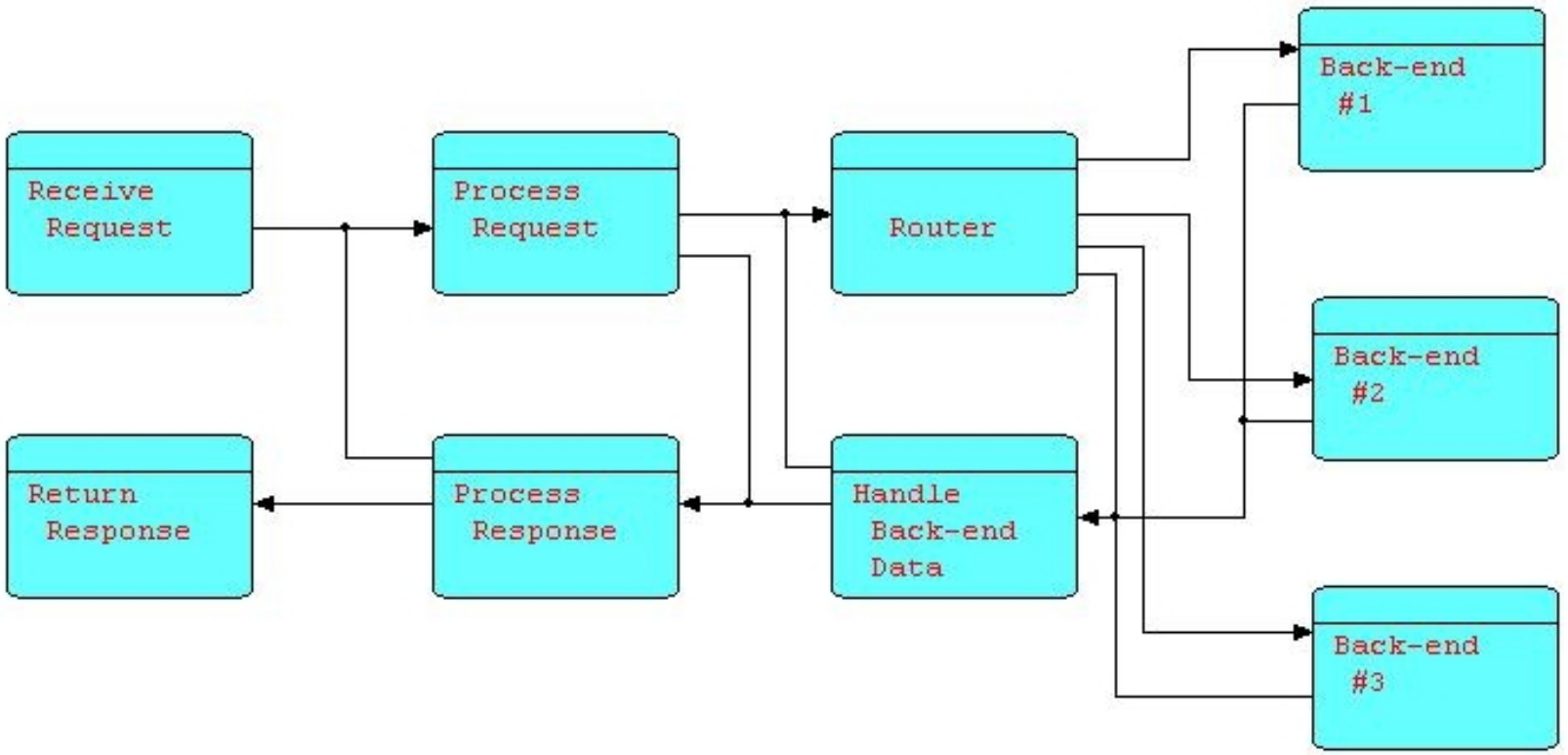
- **Finite State Machines (PDAs)**
- **Structured Analysis**
- **Step-wise decomposition**



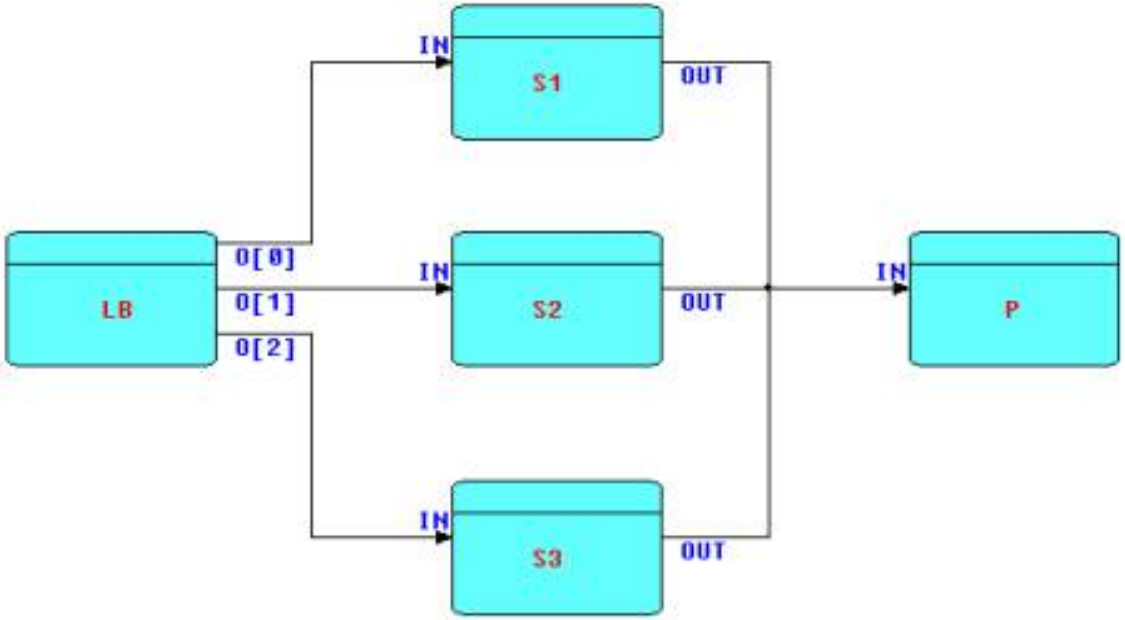
Scheduling Rules

- **Process only terminates when no more data, although it may suspend or go dormant**
- **Process can suspend on receive or send (connections are “bounded buffers”)**
- **This allows large amounts of data to be processed using finite resources**
- **Typically only one process suspended waiting for an event – not whole program**

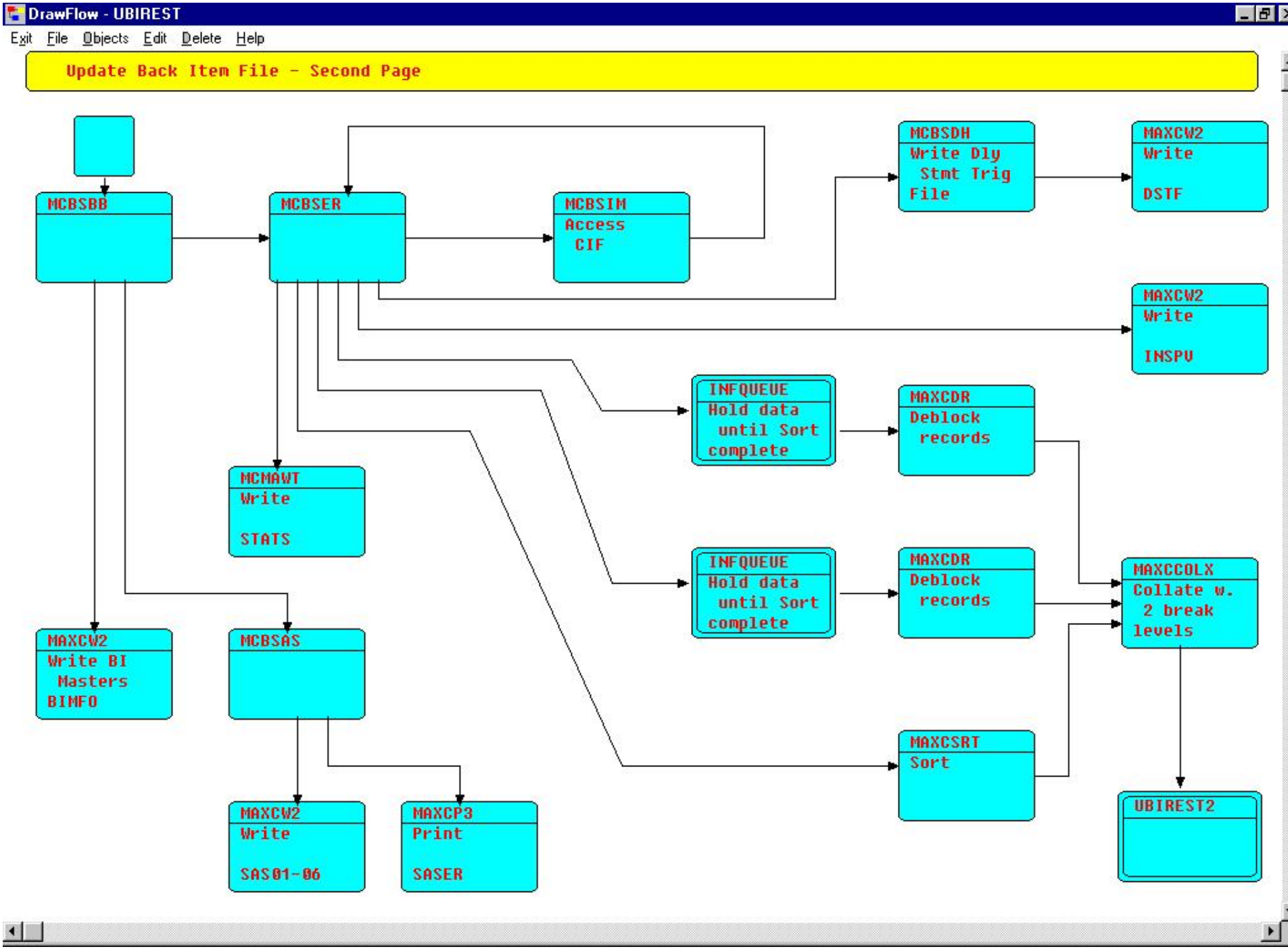
Design for an E-Brokerage Application



Load Balancing



- Sn is instance of “S”
- LB assigns on basis of least backlog
- “O[n]” is element of array port


DrawFlow - UBIREST
Exit File Objects Edit Delete Help
Update Back Item File - Second Page
MCBSBB
MCBSER
MCBSIM
Access CIF
MCBSDH
Write Dly Stmt Trig File
MAXCW2
Write DSTF
MAXCW2
Write INSPV
INFQUEUE
Hold data until Sort complete
MAXCDR
Deblock records
MCMAWT
Write STATS
INFQUEUE
Hold data until Sort complete
MAXCDR
Deblock records
MAXCCOLX
Collate w. 2 break levels
MAXCW2
Write BI Masters BIMFO
MCBSAS
MAXCSRT
Sort
MAXCW2
Write SAS01-06
MAXCP3
Print SASER
UBIREST2

Testing

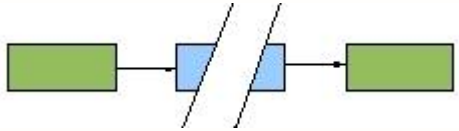
Look at data passing across connection

Test one process at a time
"Scaffolding"

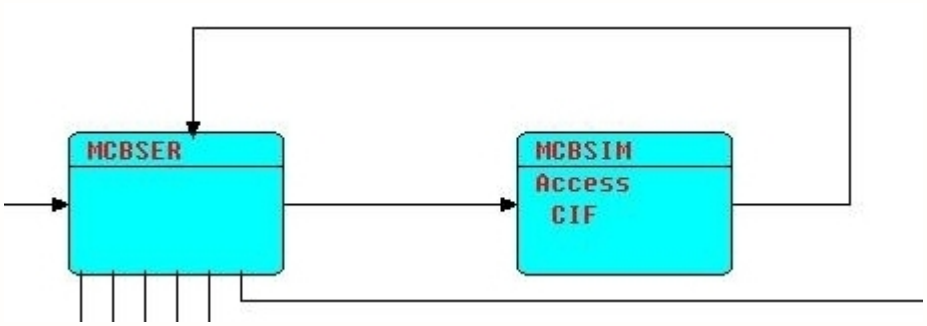
Rapid Prototyping

Go from simulation to working code

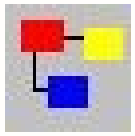
Cleave network as desired – multiple job steps, regions, machines, etc.



Simulating Subroutines

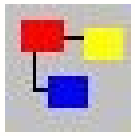


- Subroutine can be simulated using Send/Receive
- MCBSIM can also be used in “flow-through” mode
- “There is no reason we know of, however, to base an entire parallel language on this one easily programmed but not crucially important special case [calls]." Gelernter



“Active” vs. “Passive” Objects

- **FBP processes are “active objects” (*)**
- **Information packets (IPs):**
  - **have a definite lifetime**
  - **have only one owner at a time (or are on a connection)**
  - **“passive” objects**
- **5 types of “encapsulation”**
- **(*) Ellis & Gibbs (1989)**



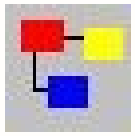
FBP Implementations

- **AMPS** z90 Assembler (IBM Canada – ca. 1970)
  (in continuous, daily use since then)
- **DFDM** Assembler / PL/I (IBM Canada, Japan)

データ・フロー・プログラミング 管理

- **THREADS** written in C (JPM)
- - just ported to C++ fibers
- **JavaFBP** Java (J. Cowan, JPM, now at 5.0)
- **C#FBP** C# (Amanda Ge, JPM)
- **CppFBP** C++ (using Boost)

- **"language-agnostic" - interface is data**



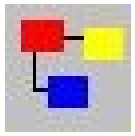
Inter-language communication

Sockets

can be used between languages on same machine, or

to communicate between different machines

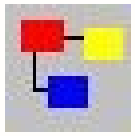
MQSeries, ActiveMQ, TIBEMS, AMQP, etc.



Data format for DrawFBP

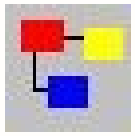
```
<net>
<title>Update</title>
<blocks>
<block>
<x> 560 </x> <y> 152 </y> <id> 6 </id>
<description>

 Print Report</description>
</block>
</blocks>
 <connections>
<connection> <fromx>108</fromx> <fromy>60</fromy>
<tox>168</tox> <toy>84</toy> <fromid>1</fromid>
<toid>3</toid>
<bends>
<bend> <x>132</x> <y> 60</y> </bend>
<bend> <x>132</x> <y> 84</y> </bend>
</bends>
</connection>
</connections>
</net>
```



Related Concepts

- MASCOT (RSRE, UK)
- Linda - Gelernter & Carriero (Yale)
- A'UM - Yoshida & Chikayama (Japan)
- BSP - Valiant
- NIL/Hermes - Rob Strom (IBM)
- “Media Objects” / KNOs - Nierstrasz etc.
- “Active Objects” - Ellis & Gibbs
- IBM’s MQSeries
- Message-driven EJBs
  - Trelliswerk, ProtoSW
  - Pervasive, Ascential (now IBM)
  - Visual Frameworks
  - etc., etc., etc.



FBP Summary

- **Flow-Based Programming: effective way to produce reliable, maintainable, large business applications**
- **Consistent view across all levels**
- **Better way to program multiprocessor computers, distributed systems**
- **Increasing interest world-wide**
- **It's taken 35+ years, but we're getting there!**

“State of the Art”