Software Development Issues for the University of La Verne

30 April 2003

Ralph W. Boaz

PILLAR CONSULTING

Software Development Issues

- "I think there is a world market for maybe five computers."
  - *Thomas Watson, Chairman of IBM, 1943*

- "Computers in the future may weigh no more than 1.5 tons."
  - *"Popular Mechanics," forecasting the relentless march of science, 1949*

Software Development Issues

What is Software Development?

Debugging The Process

Tricks of the Trade

What is it like out there?

PILLAR CONSULTING

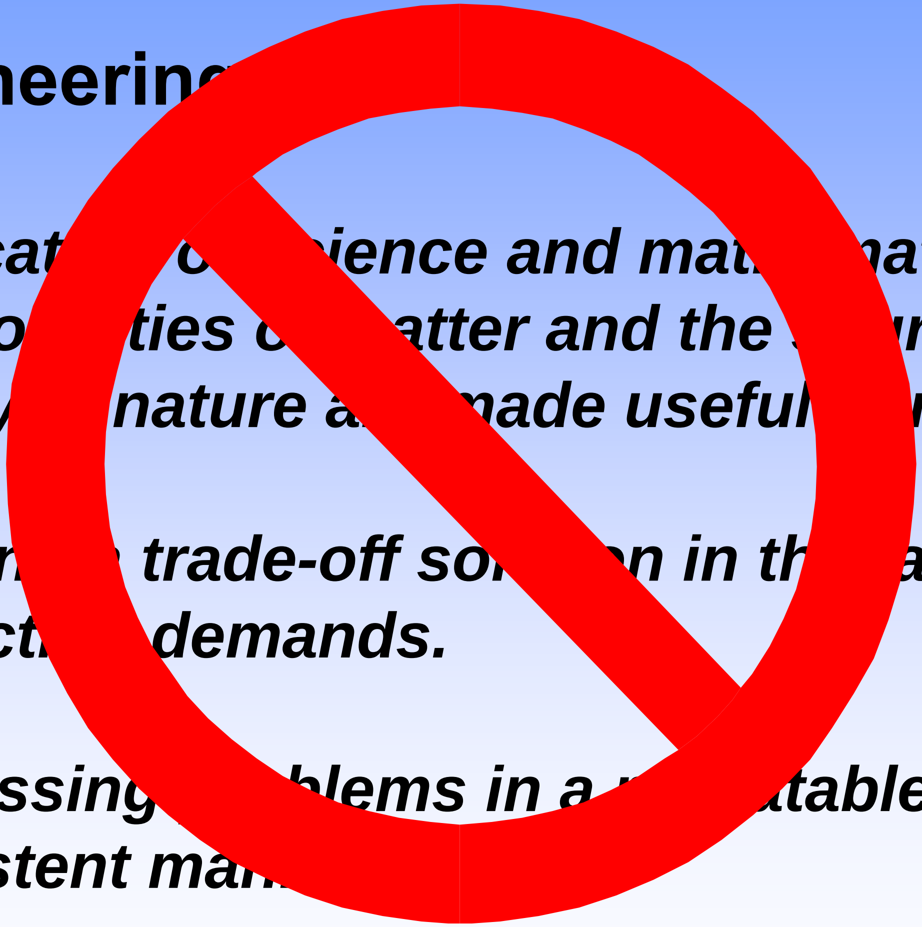
What is Software Development?

- Engineering

Application of science and mathematics by which the properties of matter and the sources of energy in nature are made useful to man.

Creating a trade-off solution in the face of conflicting demands.

Addressing problems in a repeatable and consistent manner.



PILLAR CONSULTING

What is Software Development?

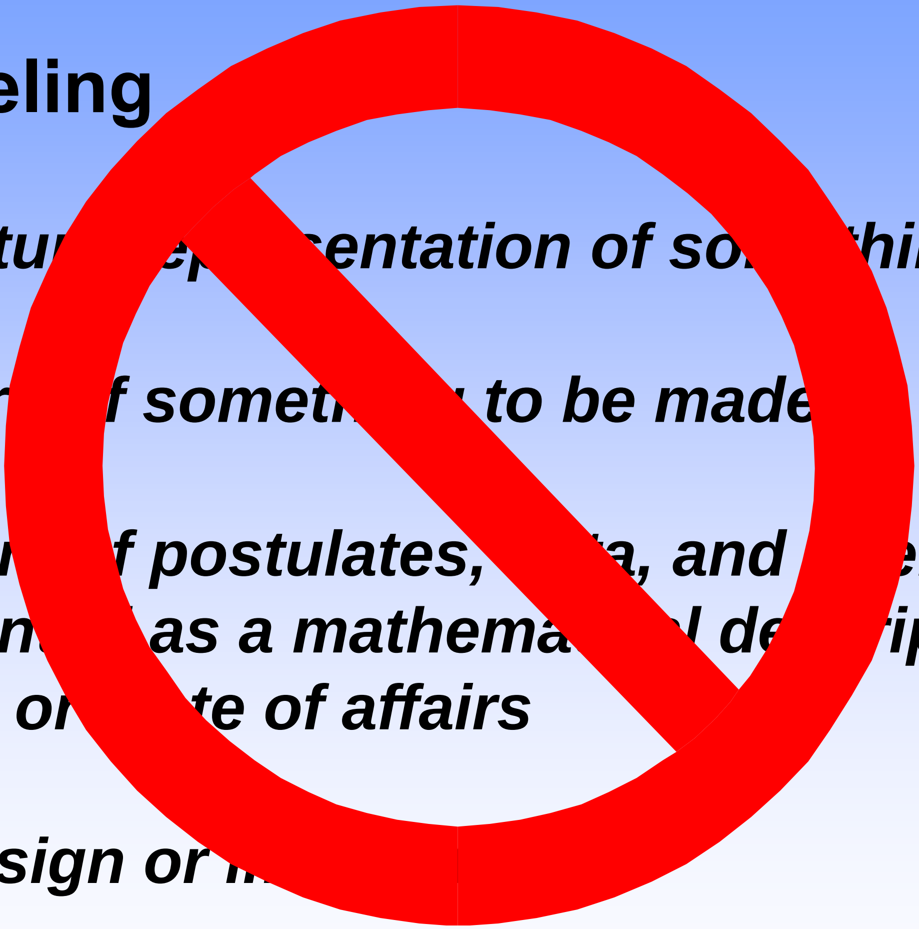
- **Modeling**

Miniature representation of something

Pattern of something to be made

System of postulates, data, and inferences presented as a mathematical description of an entity or state of affairs

To design or imitate



PILLAR CONSULTING

What is Software Development?

- It's a Game!
  - Group game
  - Goal seeking
  - Finite
  - Cooperative

- Moves of the game are made through *invention* and *communication*

What is software development?

- Historically, success came from focusing on the *invention* part of the game

- Success today will come from our skill at the *communication* part of the game

What is software development?

What are the goals of game?

- Primary goal to deliver useful, working software
- Secondary goal to leave enough “residue” to set up for the next game

Software Development Issues

What is software development?

Debugging The Process

Tricks of the Trade

What is it like out there?

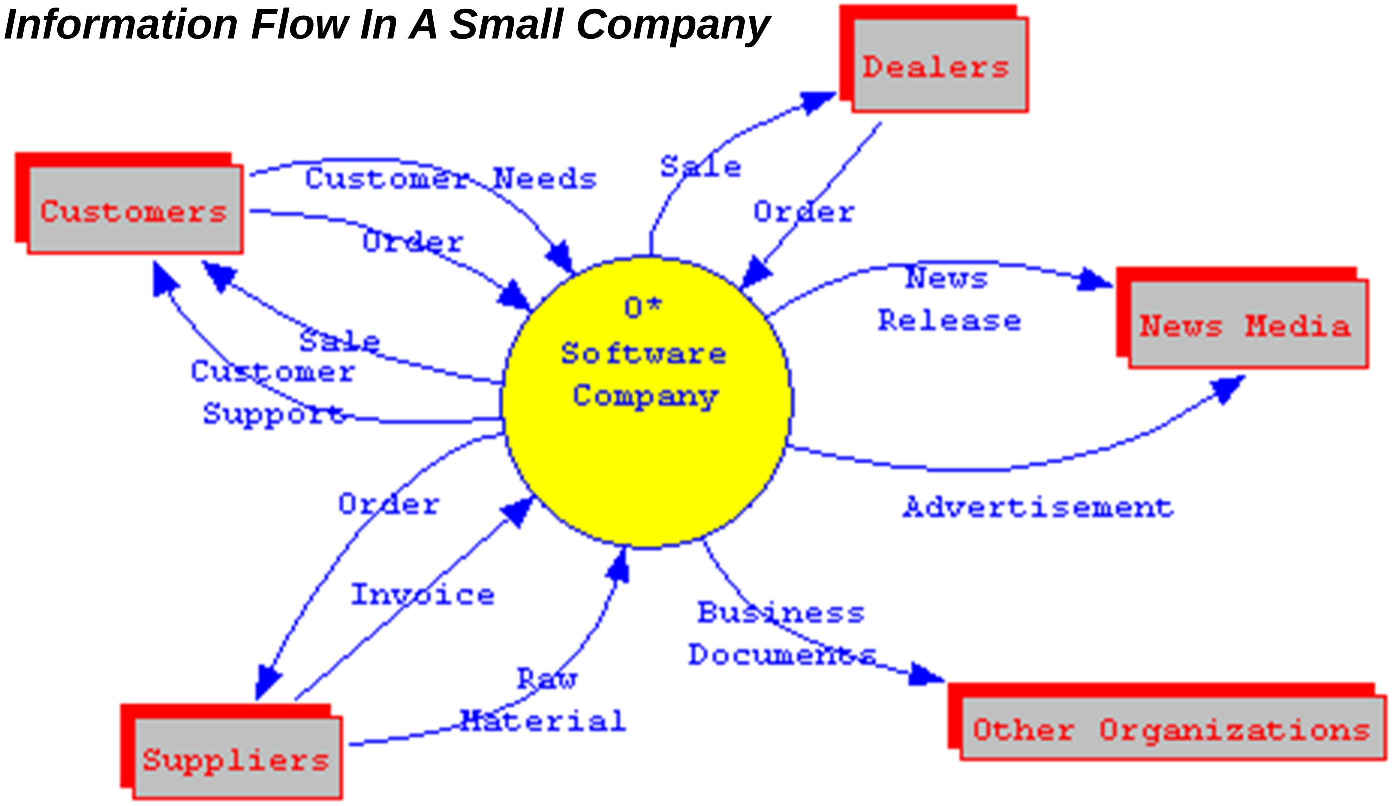
Debugging The Process

- Models
  - Serve as part of the team communication
  - Construction is only interesting as it helps us win the game
  - Are sufficient as soon as they permit the next person to move on with his work

PILLAR CONSULTING

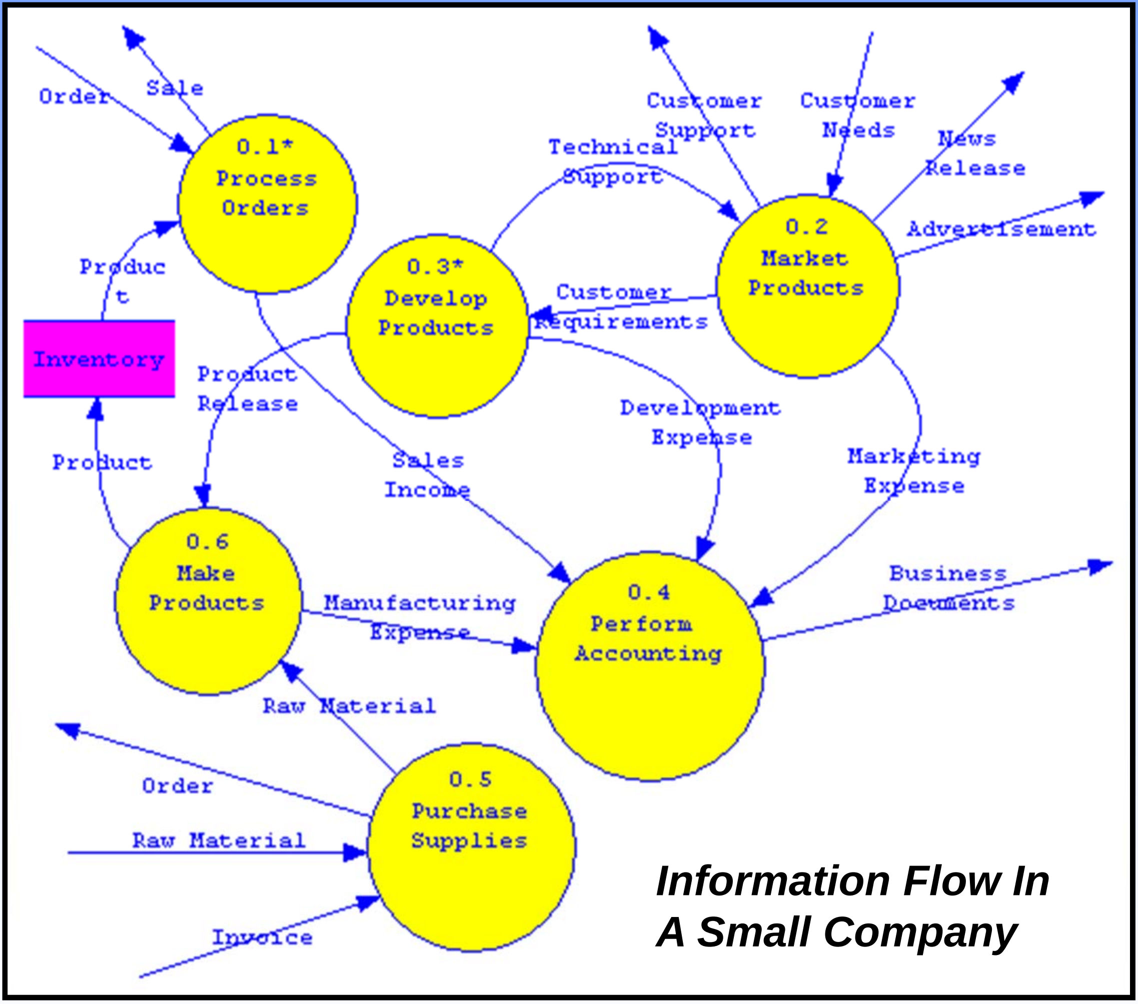
Data Flow Diagram (DFD)

Information Flow In A Small Company



PILLAR CONSULTING

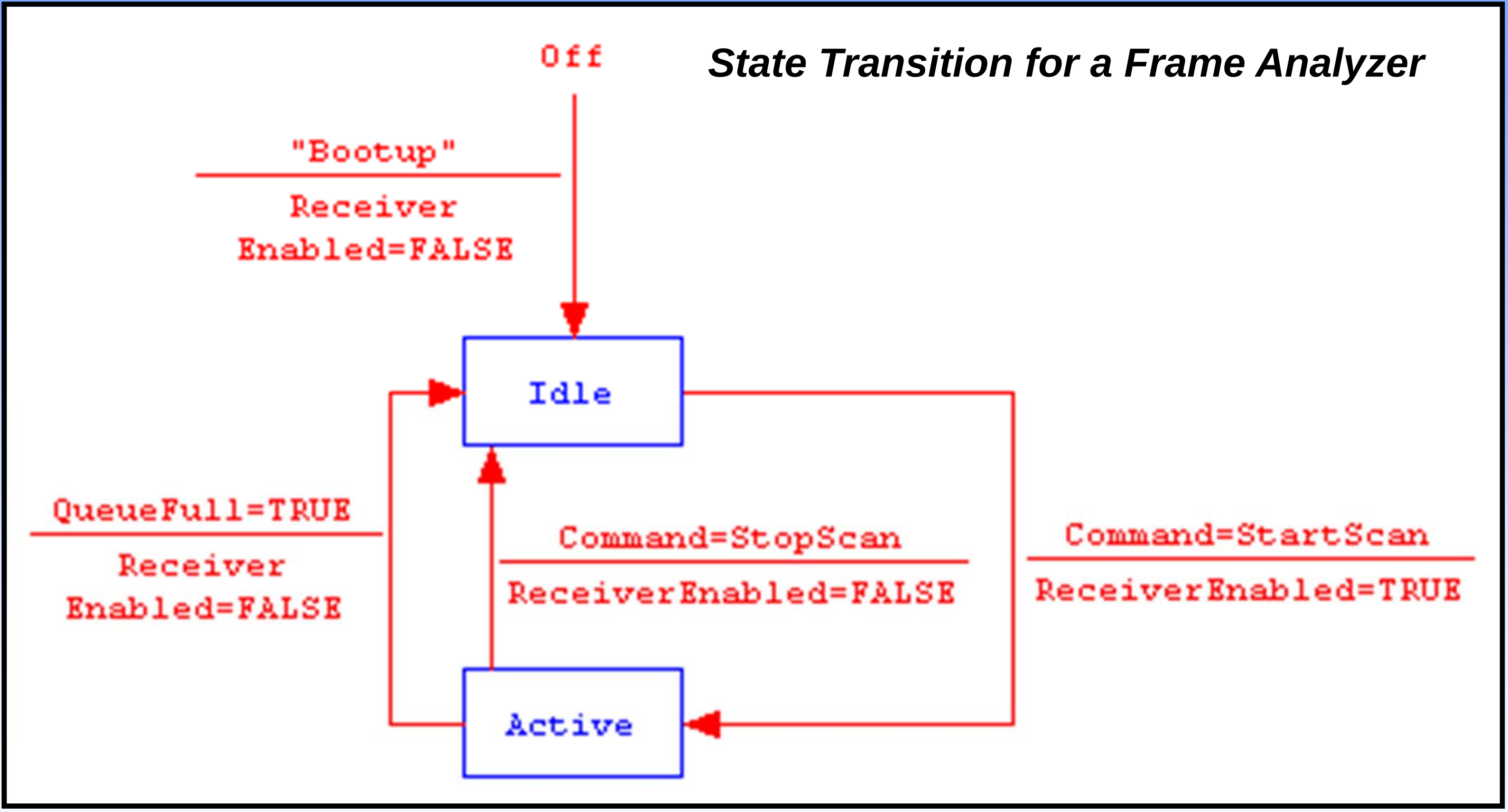
Data Flow Diagram (DFD)



*Information Flow In
A Small Company*

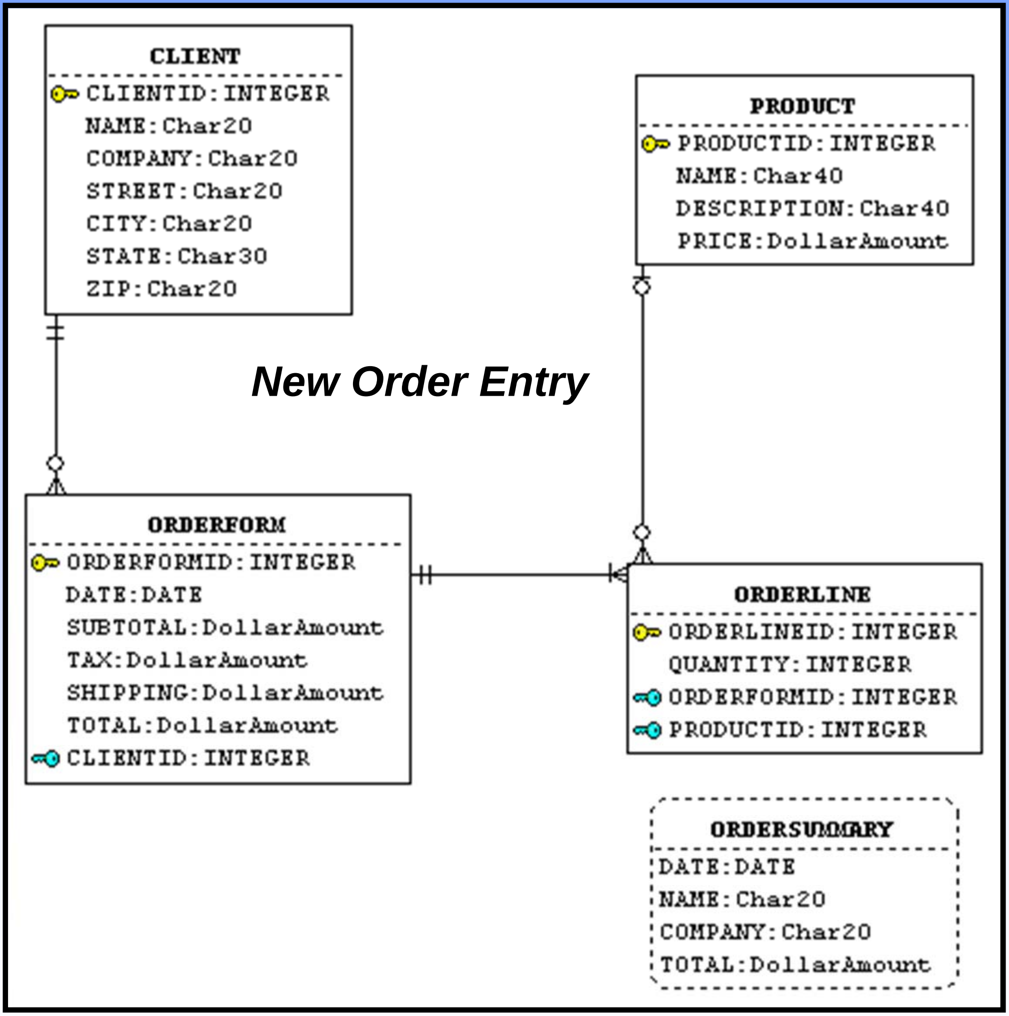
PILLAR CONSULTING

State Transition Diagram (STD)



State Transition for a Frame Analyzer

Information Engineering Notation (IEN)



New Order Entry

CLIENT
- CLIENTID:INTEGER
- NAME:Char20
- COMPANY:Char20
- STREET:Char20
- CITY:Char20
- STATE:Char30
- ZIP:Char20

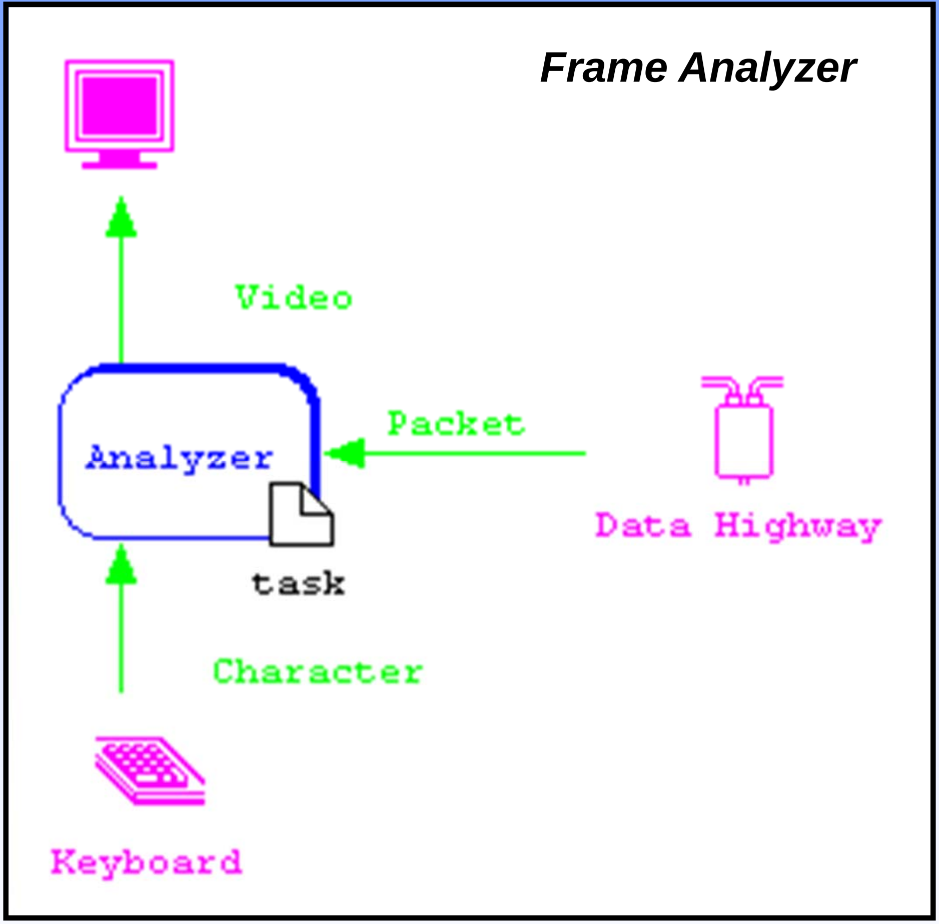
PRODUCT
- PRODUCTID:INTEGER
- NAME:Char40
- DESCRIPTION:Char40
- PRICE:DollarAmount

ORDERFORM
- ORDERFORMID:INTEGER
- DATE:DATE
- SUBTOTAL:DollarAmount
- TAX:DollarAmount
- SHIPPING:DollarAmount
- TOTAL:DollarAmount
- CLIENTID:INTEGER

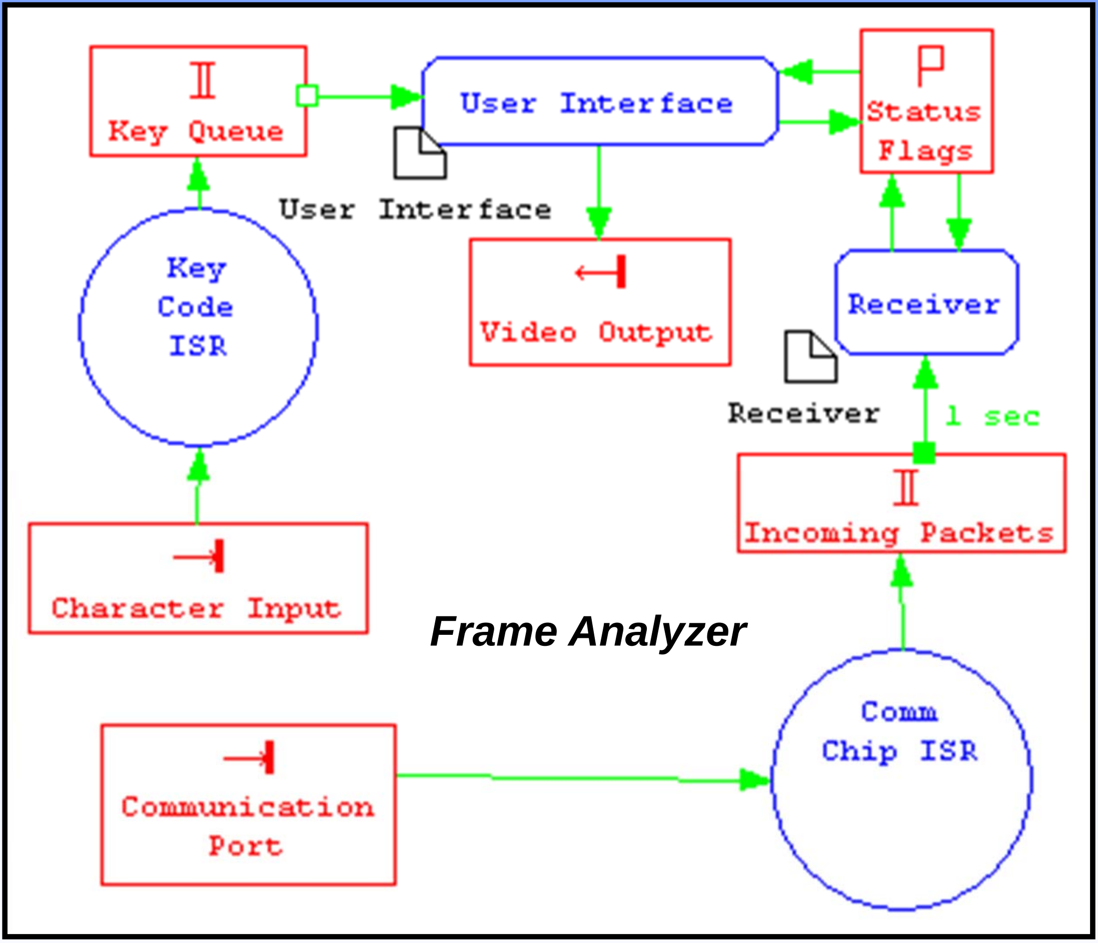
ORDERLINE
- ORDERLINEID:INTEGER
- QUANTITY:INTEGER
- ORDERFORMID:INTEGER
- PRODUCTID:INTEGER

ORDERSUMMARY
- DATE:DATE
- NAME:Char20
- COMPANY:Char20
- TOTAL:DollarAmount

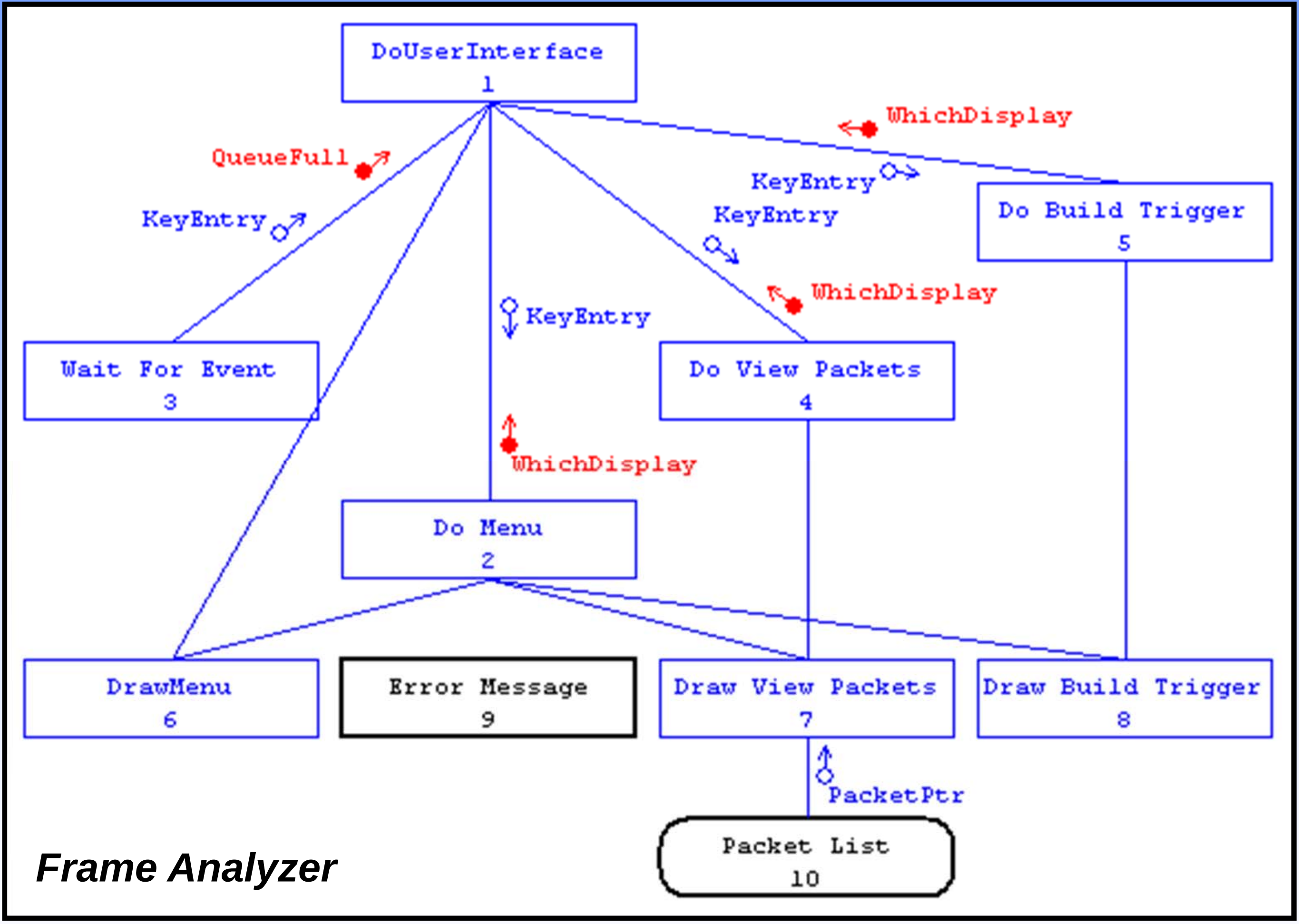
System Diagram



System Diagram

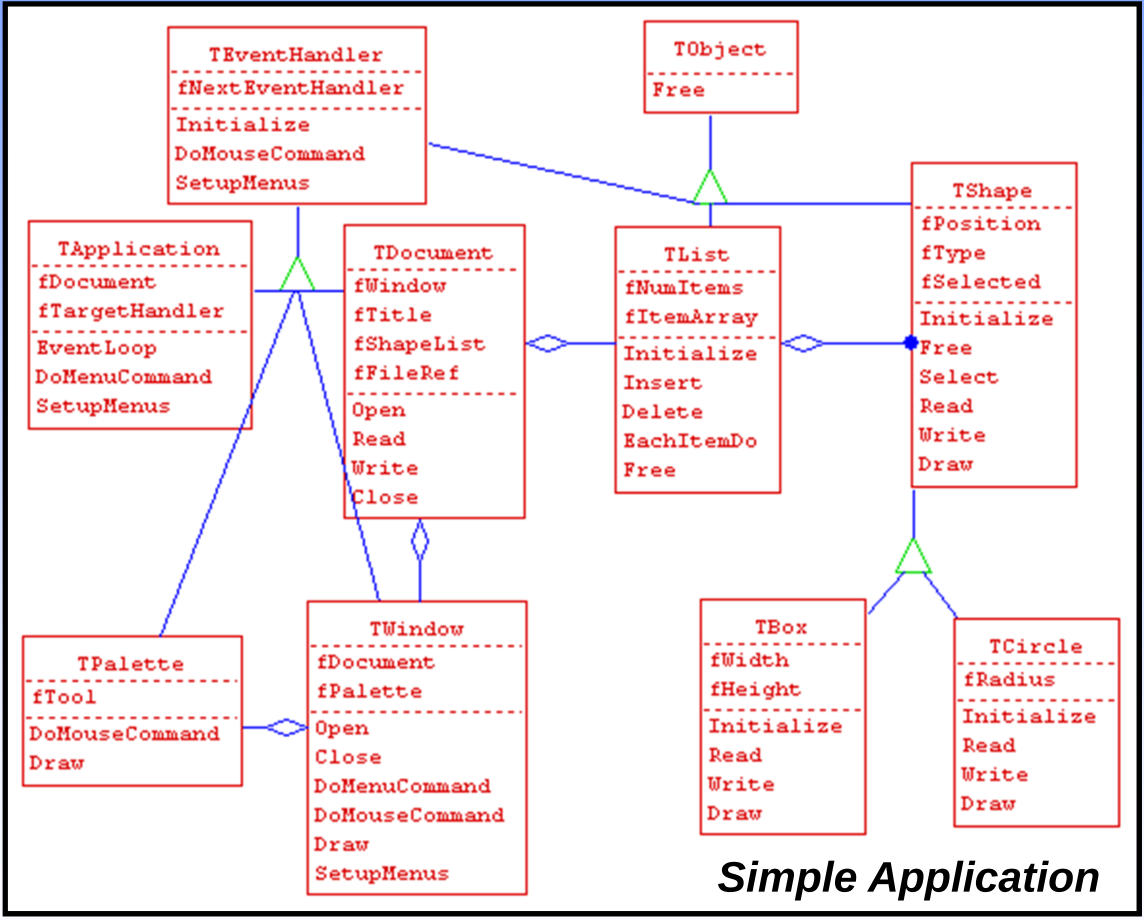


Structure Chart



PILLAR CONSULTING

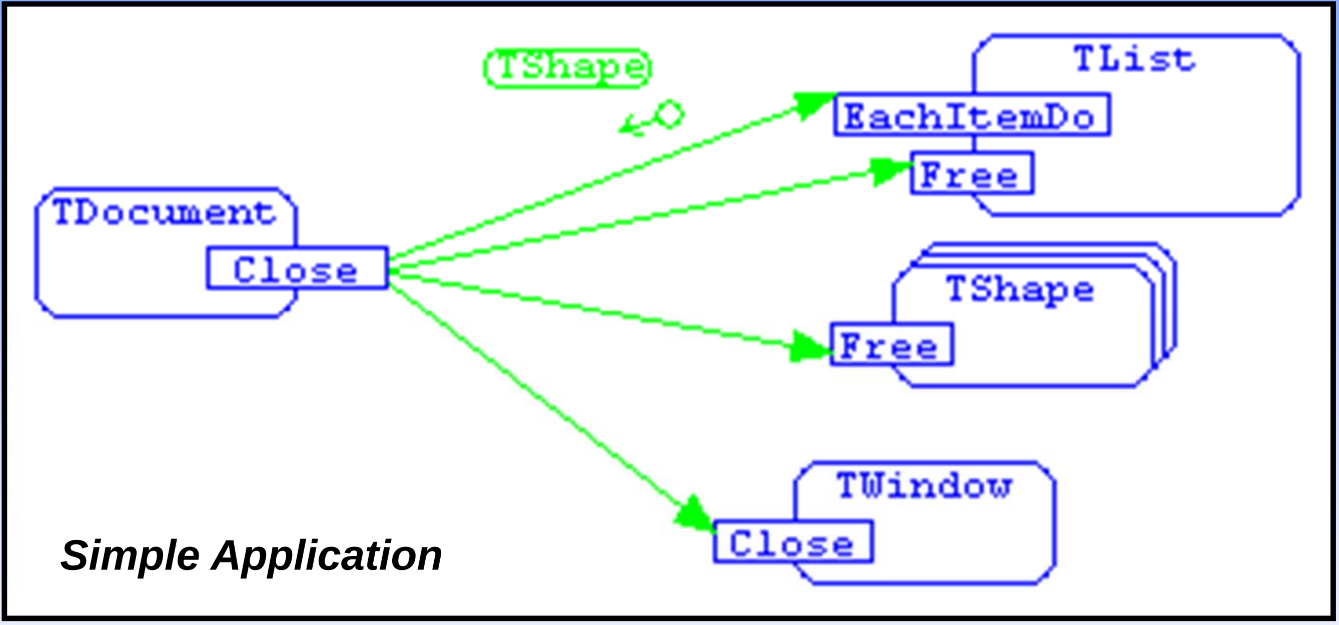
Object Modeling Technique (OMT) Class Diagram



Simple Application

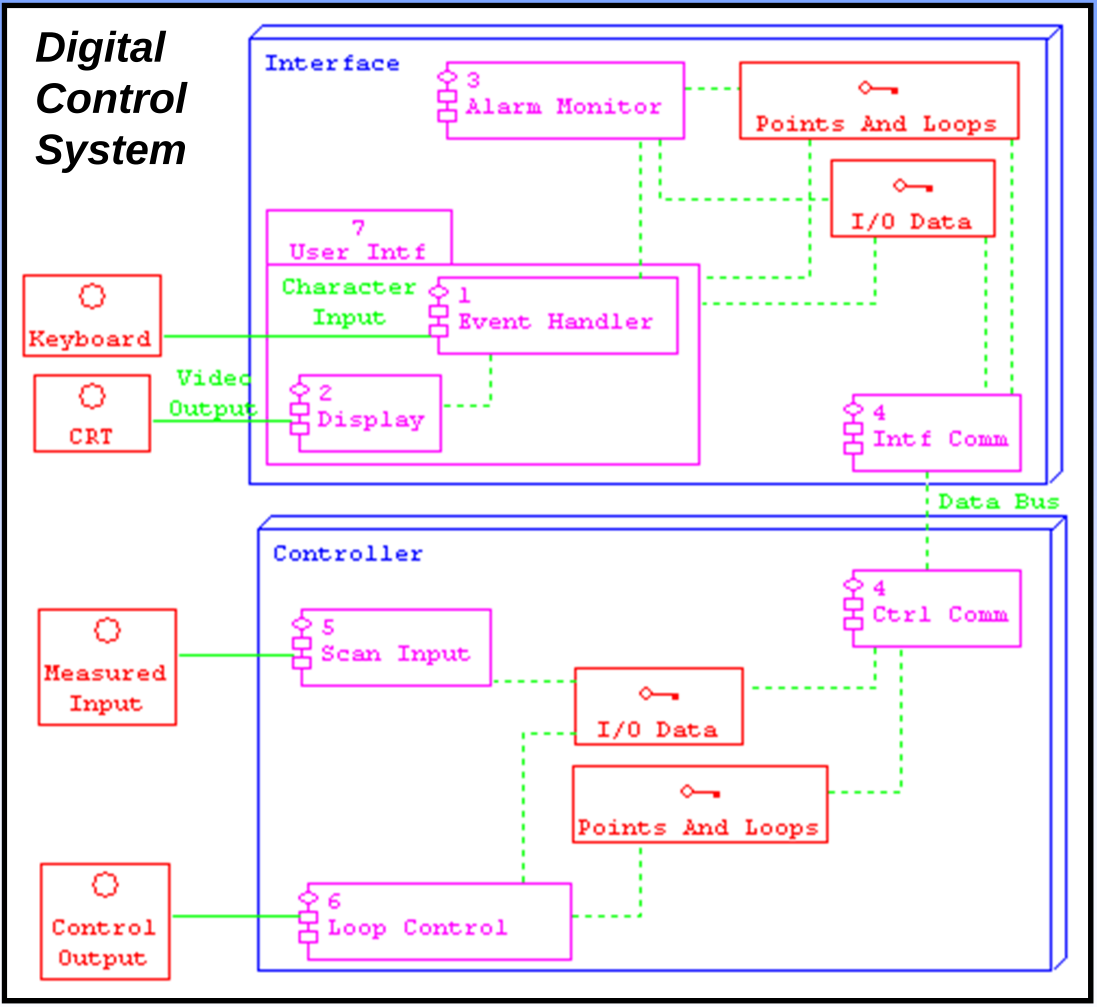
PILLAR CONSULTING

Object Modeling Technique (OMT)
Object Diagram



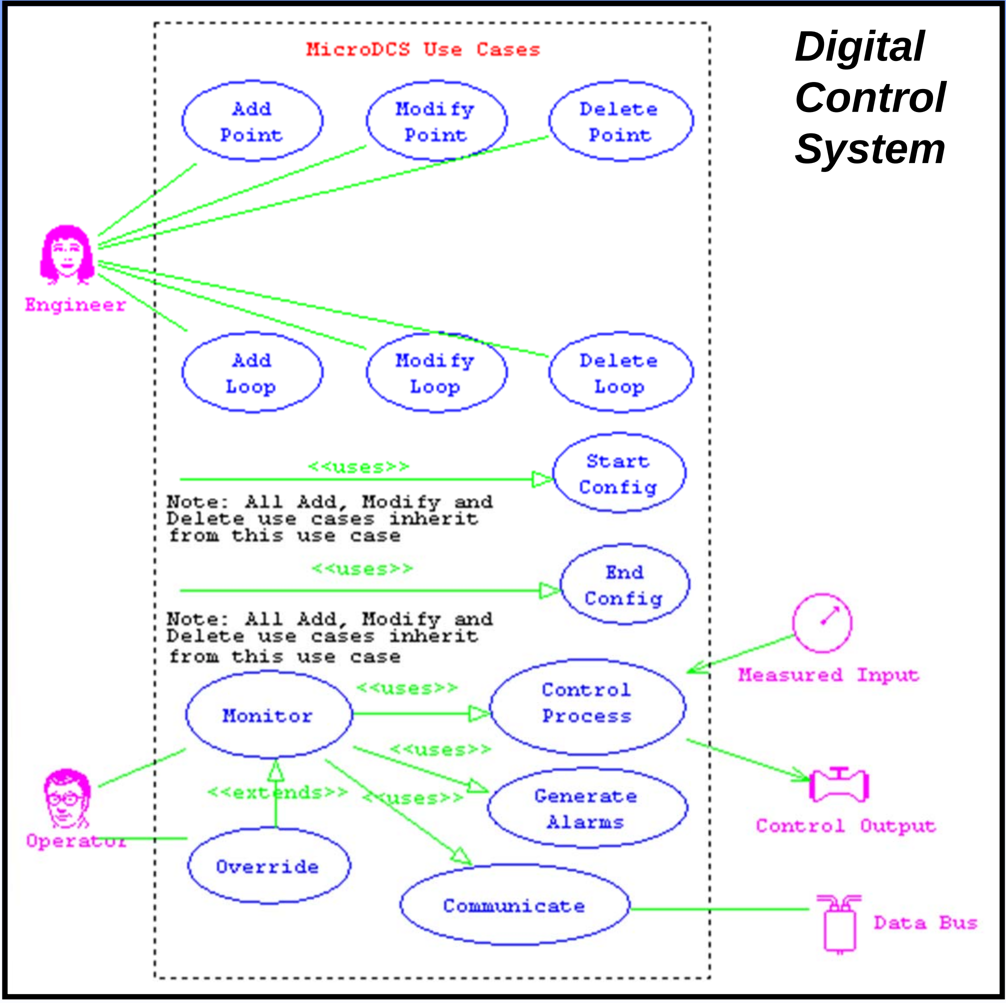
Simple Application

Unified Modeling Language (UML) System Diagram



PILLAR CONSULTING

Unified Modeling Language (UML) Use Case Diagram



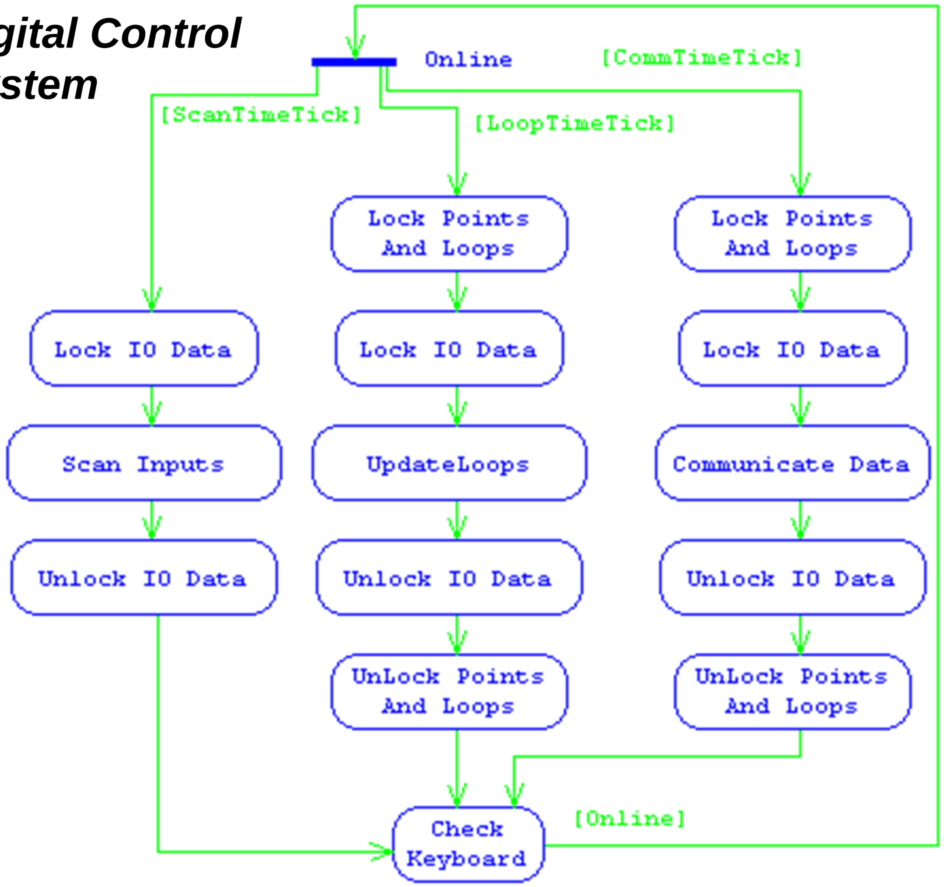
Digital Control System

Note: All Add, Modify and Delete use cases inherit from this use case

Note: All Add, Modify and Delete use cases inherit from this use case

Unified Modeling Language (UML) Activity Diagram

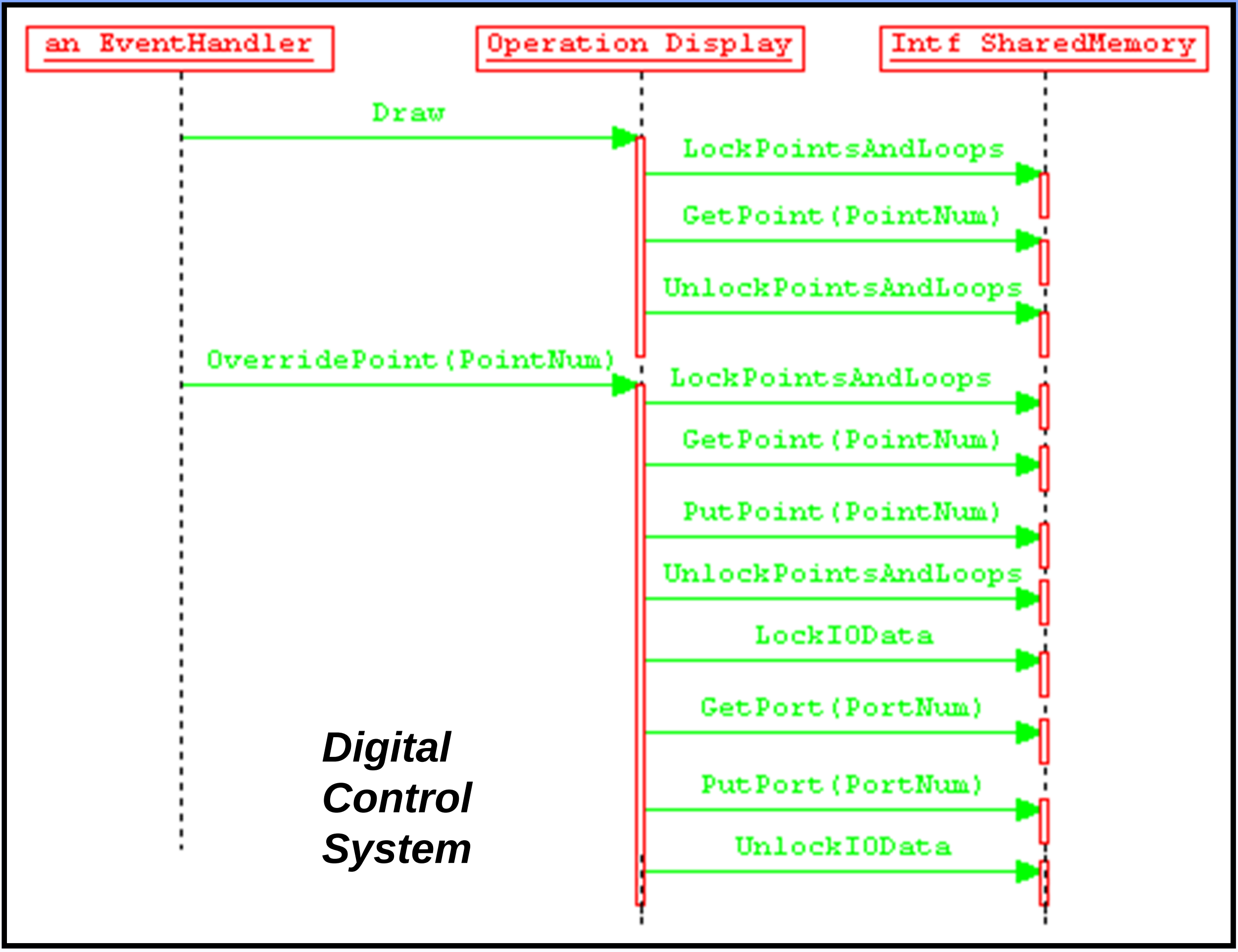
Digital Control System



Access to shared memory on the Control card is coordinated between the Scan Inputs, Loop Control and Ctrl Comm tasks during online operation.

PILLAR CONSULTING

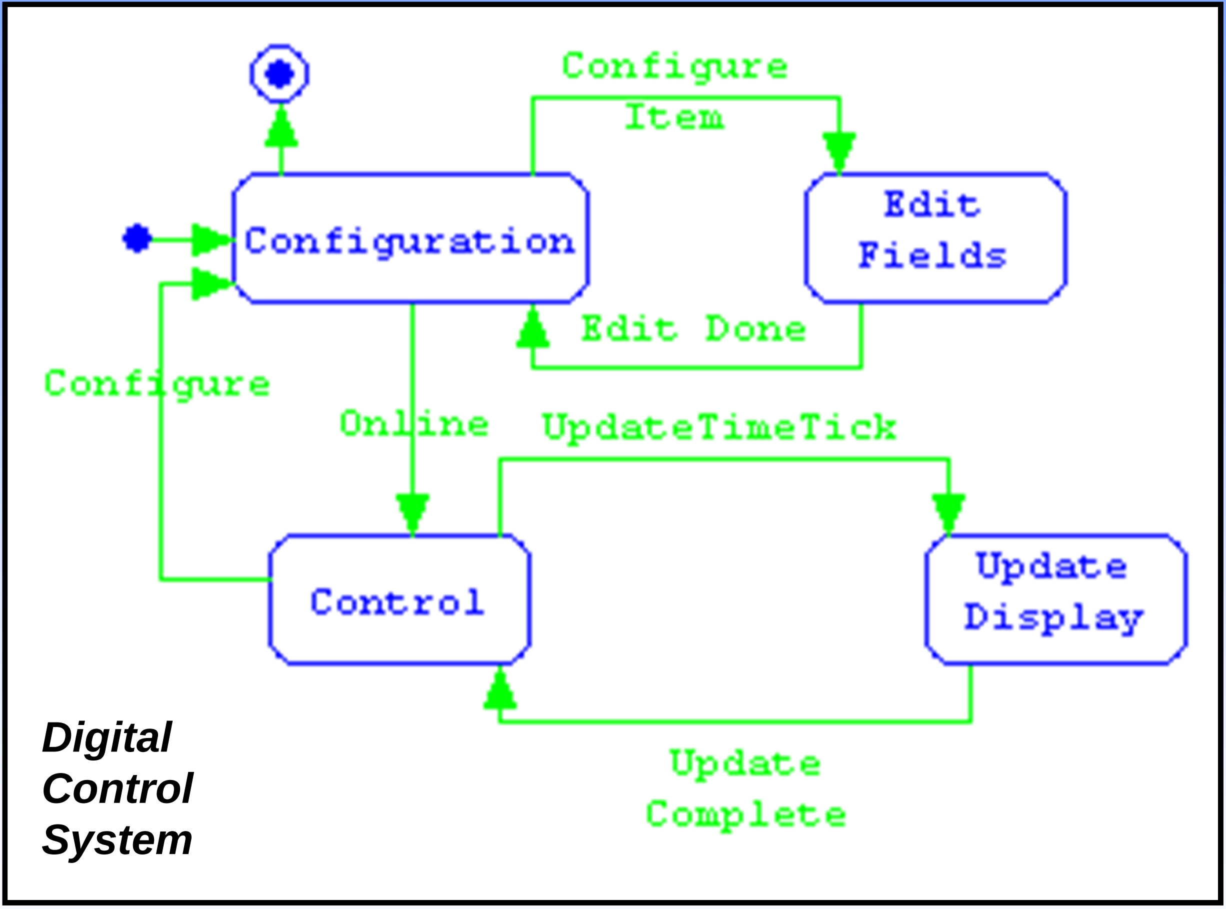
Unified Modeling Language (UML) Sequence Diagram



Digital Control System

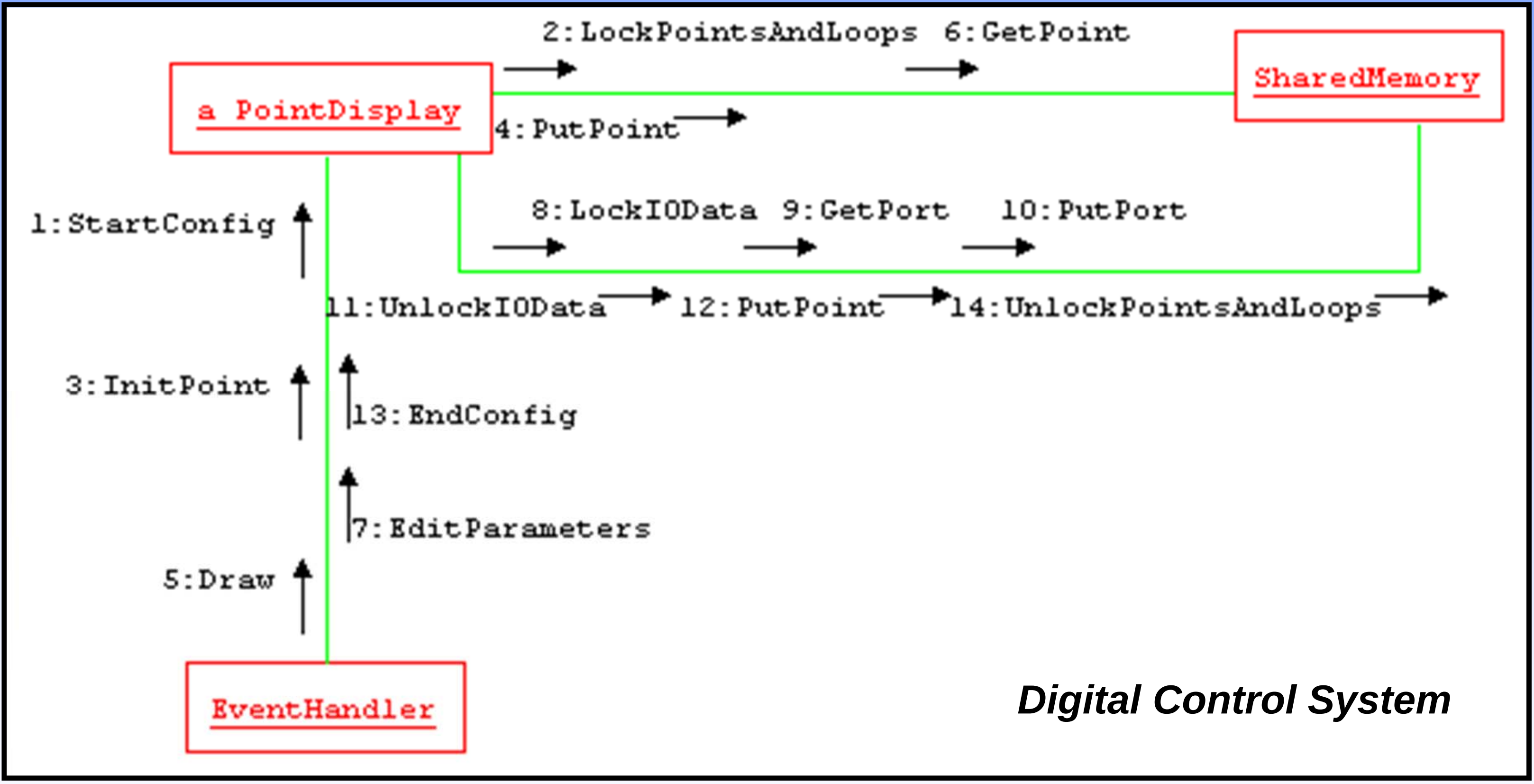
PILLAR CONSULTING

Unified Modeling Language (UML) State Chart



Digital Control System

Unified Modeling Language (UML) Collaboration Diagram



PILLAR CONSULTING

Debugging The Process

- Industry today is focused on Process

- Heavy Processes
  - Documented and formalized procedures with specific and detailed intermediate work products
  - Projects have the same documents
  - Assumes more documentation means a better product

PILLAR CONSULTING

Debugging The Process

- Agile Processes
  - Different projects need different processes
  - Light but sufficient
  - Focus on
    Skills, Communication and Community

A process is better than NO process!

Debugging The Process

- Envelope or napkin
- White board
- Paper or a CASE tool
- Code
- Paper or case tool

Software Development Issues

What is software development?

Debugging The Process

Tricks of the Trade

What is it like out there?

BAD BOYS! BAD BOYS! WHATCHA GONNA DO?
WHATCHA GONNA DO WHEN THEY COME FOR YOU?
BAD BOYS! BAD BOYS!

PILLAR CONSULTING

Tricks of the Trade

- Enable all optional compiler warnings

```
char *strcpy(char *pchTo, char *pchFrom)
{
  char *pchStart = pchTo;
  while (*pchStart = *pchFrom++)
    {}
  return (pchStart);
}
```

Tricks of the Trade

- Enable all optional compiler warnings

```
char *strcpy(char *pchTo, char *pchFrom)
{
  char *pchStart = pchTo;
  while ((*pchStart == *pchFrom++) != ‘\0’)
    {}
  return (pchStart);
}
```

Tricks of the Trade

How could we have written this fragment to detect the error automatically?

```
...
if (ch = ‘\t’)
  ExpandTab();
...
```

Is it as good as using a compiler switch?

Tricks of the Trade

- Look for ways to catch bugs automatically, with minimal effort
- Reduce the amount of programmer skill necessary to catch bugs
- Maintain both ship and debug versions of your program

Tricks of the Trade

- Strip undefined behavior from your code or use assertions to catch undefined behavior

```
void *memcpy(void *pvTo, void *pvFrom, size_t size)
{
  byte *pbTo   = (byte *)pvTo;
  byte *pbFrom = (byte *)pvFrom;

  ASSERT(pvTo != NULL && pvFrom != NULL);
  ASSERT(pvTo >= pvFrom+SIZE ||
         pvFrom >= pvTo+SIZE);

  while (size-- > 0)
    *pbTo++ = *pbFrom++;
  return (pvTo);
}
```

Tricks of the Trade

- Comment anything that isn’t clear
- Write comments that emphasize potential hazards
- Suggest a solution as long as it does not mislead people

```
/* Blocks overlap?  Use memmove. */
ASSERT(pvTo >= pvFrom+SIZE ||
       pvFrom >= pvTo+SIZE);
```

PILLAR CONSULTING

Tricks of the Trade

- Either remove implicit assumptions, or assert that they are valid
- Use assertions to detect impossible conditions
- Program defensively but don’t hide bugs
- Use a second algorithm to validate results

PILLAR CONSULTING

Tricks of the Trade

- Don’t wait for bugs to happen - use startup checks
- Testers are not there for testing your code
- Test code is forever
- Not works of art

Tricks of the Trade

- Fortify your subsystems

- What’s a subsystem?
  - Hides its implementation details
  - Has a complexity of its own
  - Provides a few key entry point

Tricks of the Trade

- The two most important questions after writing a subsystem

1) How are programmers going to misuse this subsystem?

2) How can I detect these problems automatically?

Tricks of the Trade

- Step through your code

- Don’t wait until you have a bug to step through your code

- Step through every code path
  - Operators &&, ||, and ?: have two paths
  - Use the debugger to display each side of the expression

- Focus on data flow

Tricks of the Trade

- Make it hard to ignore error conditions
  - don’t bury error codes in return values

From Kernighan and Richie -

```
...
char c;

c = getchar();
if (c == EOF)
...
```

getchar gets an int, of course

Tricks of the Trade

- Make it hard to ignore error conditions
  - don’t bury error codes in return values

```
flag fGetChar(char *pch);    /* prototype */

if (fGetChar(&ch))
  successful -- ch has the next character
else
  hit EOF -- ch is garbage
```

Tricks of the Trade

- Don’t write multipurpose functions
  - Write separate functions to allow stronger argument validation
- Make the code intelligible at the point of call

Tricks of the Trade

- **Avoid boolean arguments**

Bad

```
UnsignedToStr(u, str, TRUE)
UnsignedToStr(u, str, FALSE)

/* blah, blah, blah.  If fDecimal is TRUE,
   blah, blah, blah.  If fDecimal is FALSE,
   blah, blah, blah... */
void UnsignedToStr(unsigned u, char *strResult,
                   flag fDecimal);
```

Tricks of the Trade

Better

```
#define BASE10 1
#define BASE16 0

UnsignedToStr(u, str, BASE10)
UnsignedToStr(u, str, BASE16)
```

Best

```
void UnsignedToStr(unsigned u, char *str,
                   unsigned base);
```

Tricks of the Trade

- **Handle your special cases just once**

```
void IntToStr(int i, char *str)
{
  int iOriginal = i;
  char *pch;
  if (iOriginal < 0)
    i = -i;
  pch = str;
  do
    *pch++ = (i % 10) + ‘0’;
  while ((i /= 10) > 0);
  if (iOriginal < 0)
    *pch++ = ‘-’;
  *pch = ‘\0’;
  ReverseStr(str);
}
```

Tricks of the Trade

- Establish priorities and stick to them

RAY'S LIST	SETA'S LIST
Correctness	Correctness
Global efficiency	Testability
Size	Global efficiency
Local efficiency	Maintainability
Personal convenience	Consistency
Maintainability	Size
Personal expression	Local efficiency
Testability	Personal expression
Consistency	Personal convenience

Software Development Issues

What is software development?

Debugging The Process

Tricks of the Trade

What is it like out there?

What is it like out there?

- Find a mentor
- Play by the rules
- There are other games going on
- Domain knowledge is good
- Get a variety of experiences
- Keep your skills current

PILLAR CONSULTING

Software Development Issues

[END PROGRAM]

PILLAR CONSULTING