Structured Analysis & Structured Design

Presentation · February 1997

DOI: 10.13140/RG.2.2.18323.50725

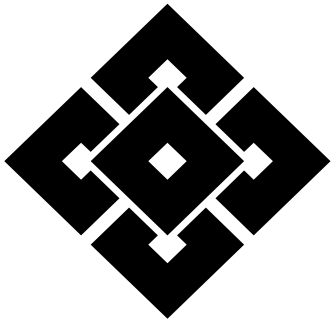
1 author:

James Cusick
IEEE Senior Member
155 PUBLICATIONS **434** CITATIONS

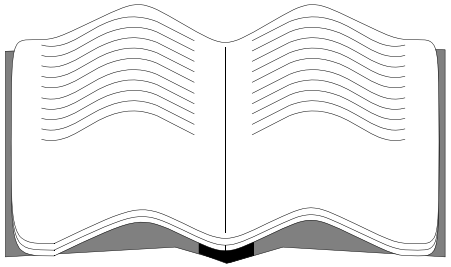
SEE PROFILE

STRUCTURED ANALYSIS &

STRUCTURED DESIGN



READING COMMENTS OR QUESTIONS



THINKING ABOUT SYSTEMS ANALYSIS

EARLY LIFE CYCLE NEEDS FOR ANALYSIS

Feasibility: defining preferred concept and its superiority to alternative concepts

Requirements: complete, validated specifications of required functions, interfaces, and performance.

SYSTEMS ANALYSIS TASKS

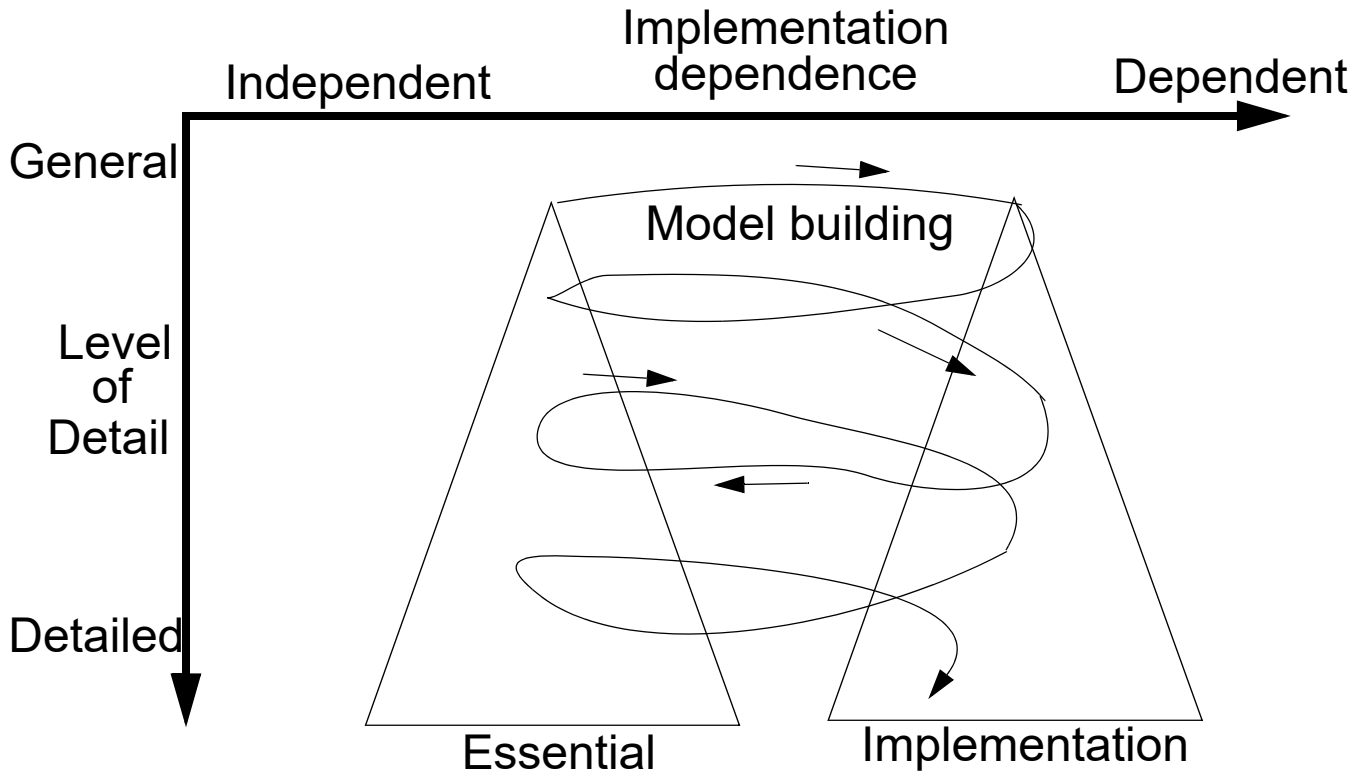
1. Problem Recognition
2. Evaluation & Synthesis
3. Modeling
4. Specification & Architecture
5. Review

ESSENTIAL vs. IMPLEMENTATION MODELS

Essential Model:

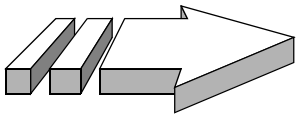
- Perfect Solution
- No Technical Limitations
- Work Towards Reality

ESSENTIAL AND IMPLEMENTAION MODELS



CONCEPTUAL MODEL SERIES

1. High level user communication model
   State Problem and Goals
2. System Solution Model
3. High Level Design Model
4. Control Structure of Software
5. Data Flow Representation
6. Unit Level Model



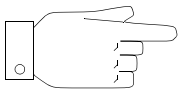
THEN CODE IT!

The knot is composed of three wraps situated one below the other along the axis of the post. The wraps enclose a segment of rope used to form the knot, which lies parallel to the axis of the post. This segment loops around the bottom wrap and ascends vertically on the outside of the wraps to form the beginning of the top wrap.

LET'S SEE YOUR DRAWING!!!

TRAINING OF A SYSTEMS ANALYST

- ☑ Systems Approach
- ☑ Defining Problems
- ☑ Data Gathering
- ☑ Interviewing Techniques
- ☑ Qualitative Data Analysis
- ☑ Quantitative Data Analysis



TARGET BUSINESS EXPERIENCE

TRAINING OF A SYSTEMS ANALYST

- Controlling data I/O
- DP Controls
- Security
- Documentation
- Logical Systems Review
- Economic Systems Review
- Hardware
- Software
- Data Communication Concepts
- Data Base Concepts
- File Organization Concepts
- Selling Systems

TRAINING OF A SYSTEMS ANALYST

- Work Measurement
- Work Simplification
- Manuals
- Effective Writing
- Indexing & Coding
- Forms Design
- Office Machines

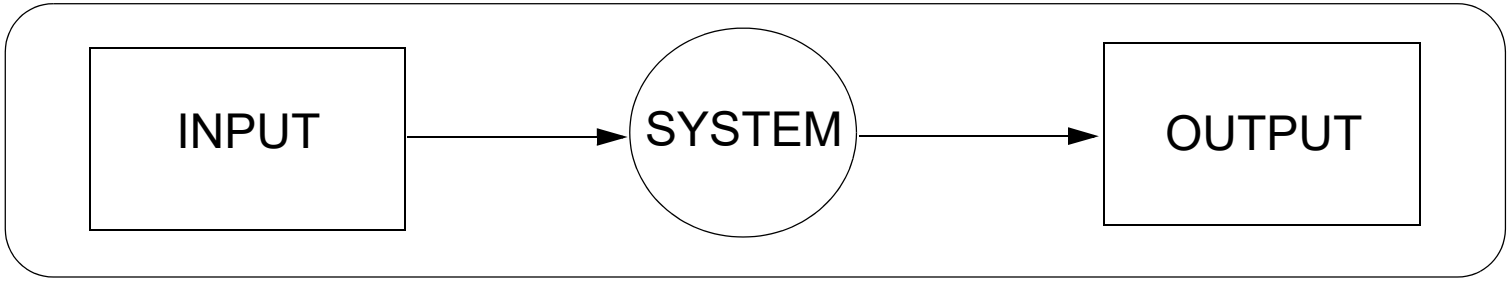
STRUCTURED ANALYSIS TECHNIQUES

STRUCTURED ANALYSIS

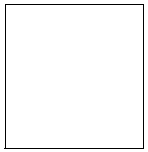
A METHOD OF PRODUCING STRUCTURED SPECIFICATIONS

* GRAPHIC AND CONCISE
* TOP-DOWN PARTITIONED
* NON-REDUNDANT
* ESSENTIAL

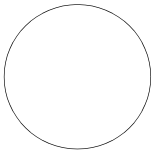
COMPUTER SYSTEMS SEEN AS INFORMATION TRANSFORM
Tools: DFD, Data Dictionary, STD, E-R Diagram



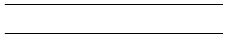
DATA FLOW DIAGRAM NOTATIONAL BASICS



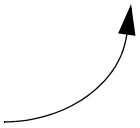
EXTERNAL ENTITY



PROCESS



DATA STORE



DATA ITEM

STATE TRANSITION DIAGRAM

* MODEL OF FINITE STATE BEHAVIOR
* IDENTIFY WHEN TRANSFORMS ARE TRIGGERED

STATE TRANSITION DIAGRAM NOTATIONAL BASICS

SYSTEM STATE

TRANSITION FROM STATE TO STATE

condition
action

AS ASSOCIATED WITH TRANSITION

A PROBLEM TO SOLVE

PROBLEM STATEMENT

METRO MOTOR VEHICLE TRAFFIC PROBLEMS CAUSE ECONOMIC DECLINE AND REDUCTION IN QUALITY OF LIFE FOR RESIDENTS.

INTEGRATED TRAFFIC CONTROL SYSTEM IS PLANNED.

YOUR ROLE IS SOFTWARE CONTRACTOR FOR SUBSYSTEM OF THIS SOLUTION.

PROBLEM ANALYSIS

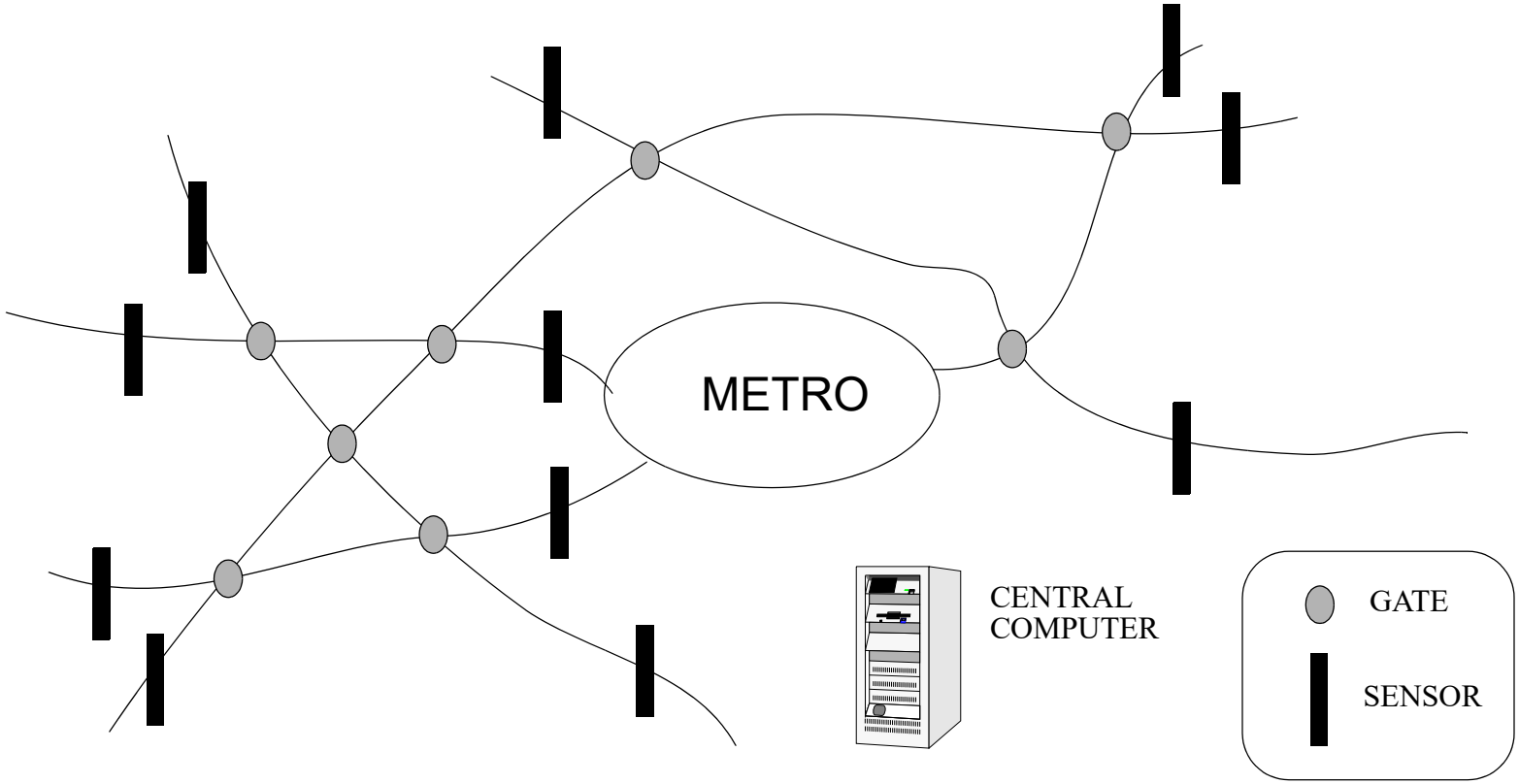
* Lack of integrated traffic and road access management system.

* Knowing both traffic volume and road conditions, coupled with ability to shift traffic from one road to another needed.

* Sensors will report on traffic flows on real time basis thus supporting road access determination.

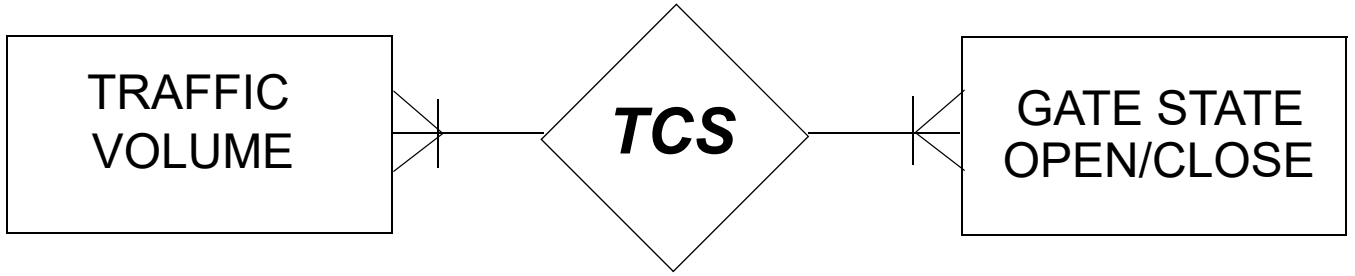
* Provide Analysis & Design Solution for Sensor

TCS (Traffic Control System) MODEL

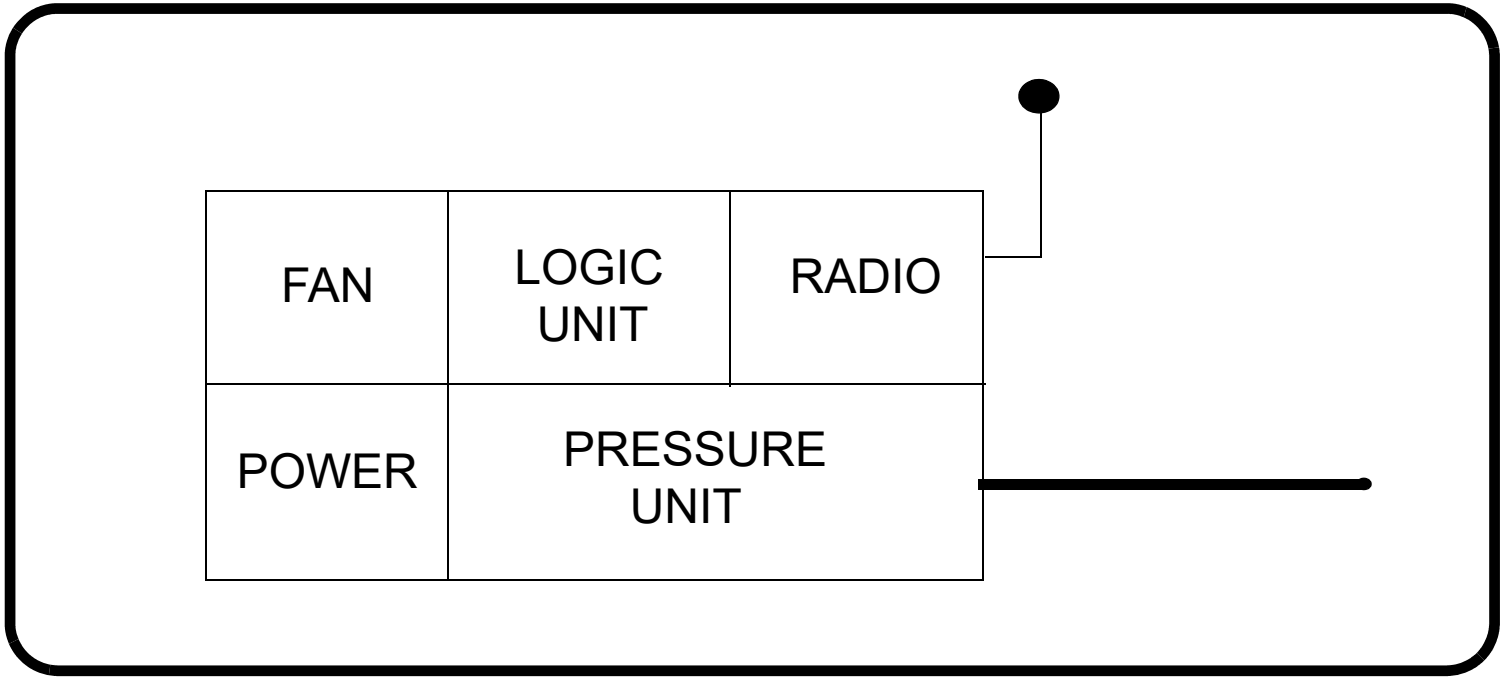


TCS HIGH LEVEL ARCHITECTURE

ENTITY RELATIONSHIP DIAGRAM: E-R



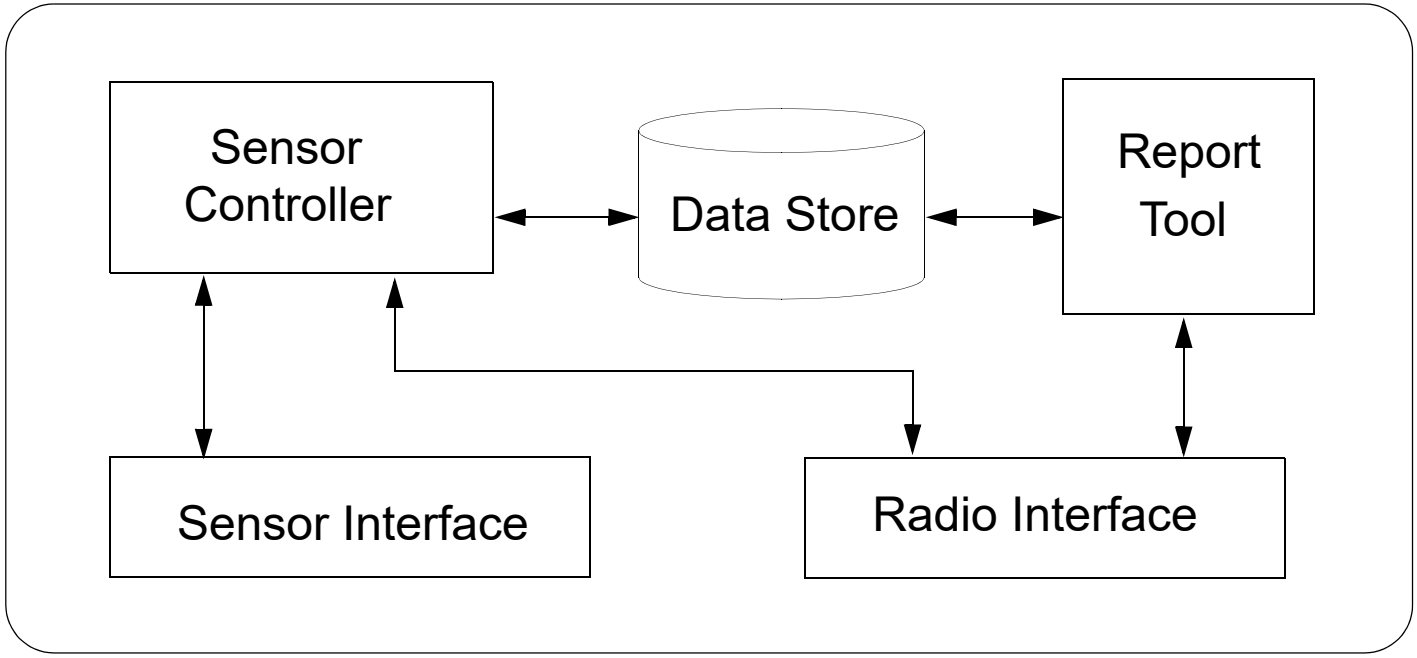
TRAFFIC SENSOR SYSTEM DESIGN



TRAFFIC SENSOR REQUIREMENTS

1. Control sensor and radio
2. Receive commands from central system
3. Collect local traffic data
4. Return traffic reports including:
   weight, speed, volume, frequency
5. Handle abnormal conditions, i.e.,
   simultaneous vehicles, multiple axles, etc.

TRAFFIC SENSOR SOFTWARE ARCHITECTURE



TRAFFIC SENSOR SOFTWARE FUNCTIONALITY

SENSOR CONTROLLER

- Poll each device
- Load operational commands
- Execute control of sensor and radio
- Update data store

SENSOR DATA STORE

- Store operational parameters (sampling frequency, location)
- Store report types
- Store sensor data

TRAFFIC SENSOR SOFTWARE FUNCTIONALITY

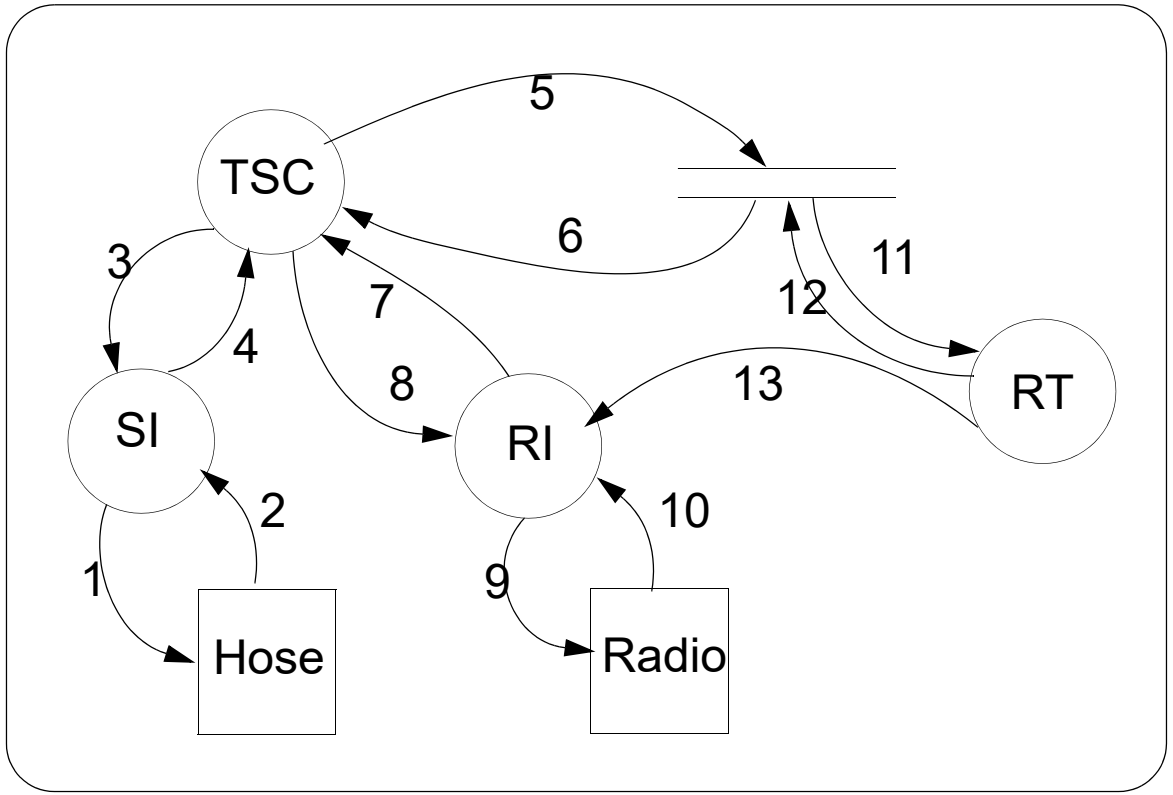
REPORT TOOL

- Format data
- Upload data
- Respond to format changes

DEVICE INTERFACES

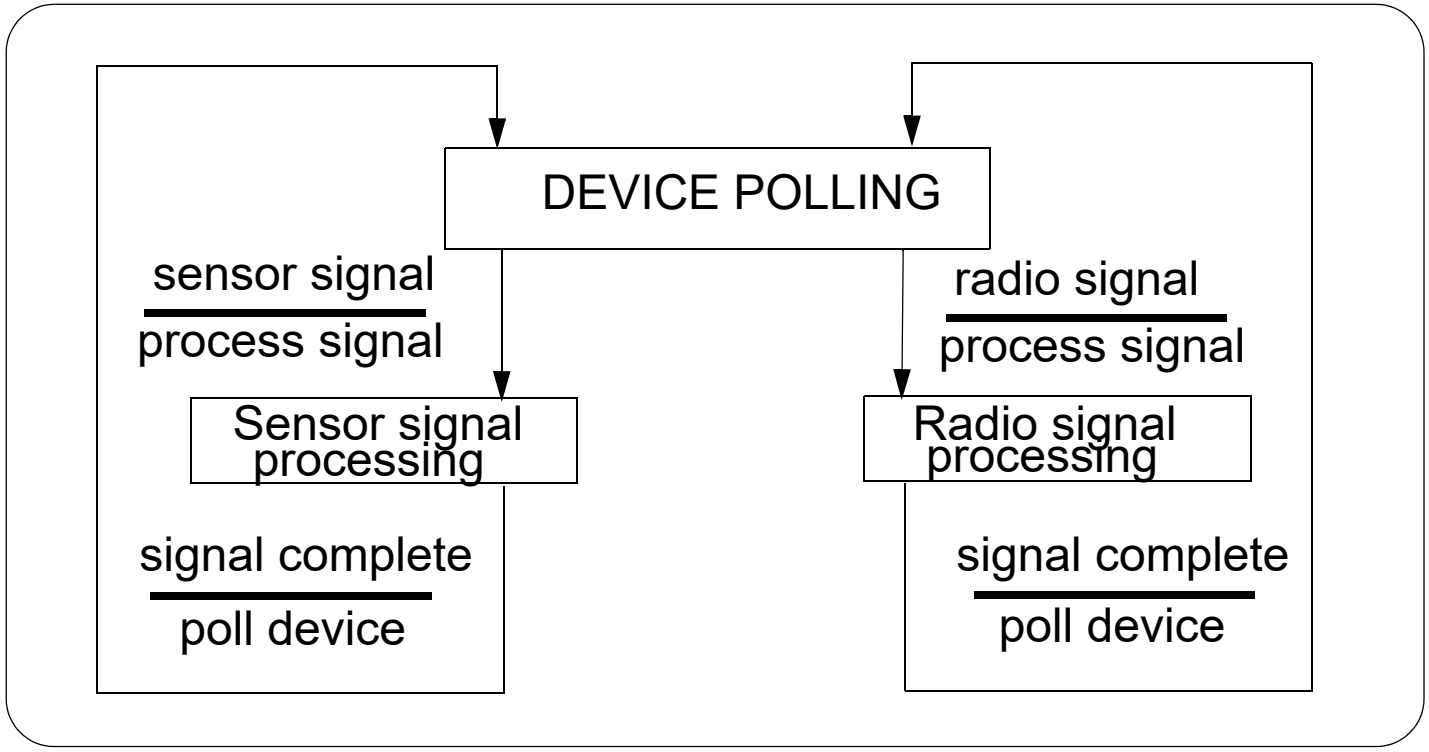
- Send & receive commands

TRAFFIC SENSOR DATA FLOW

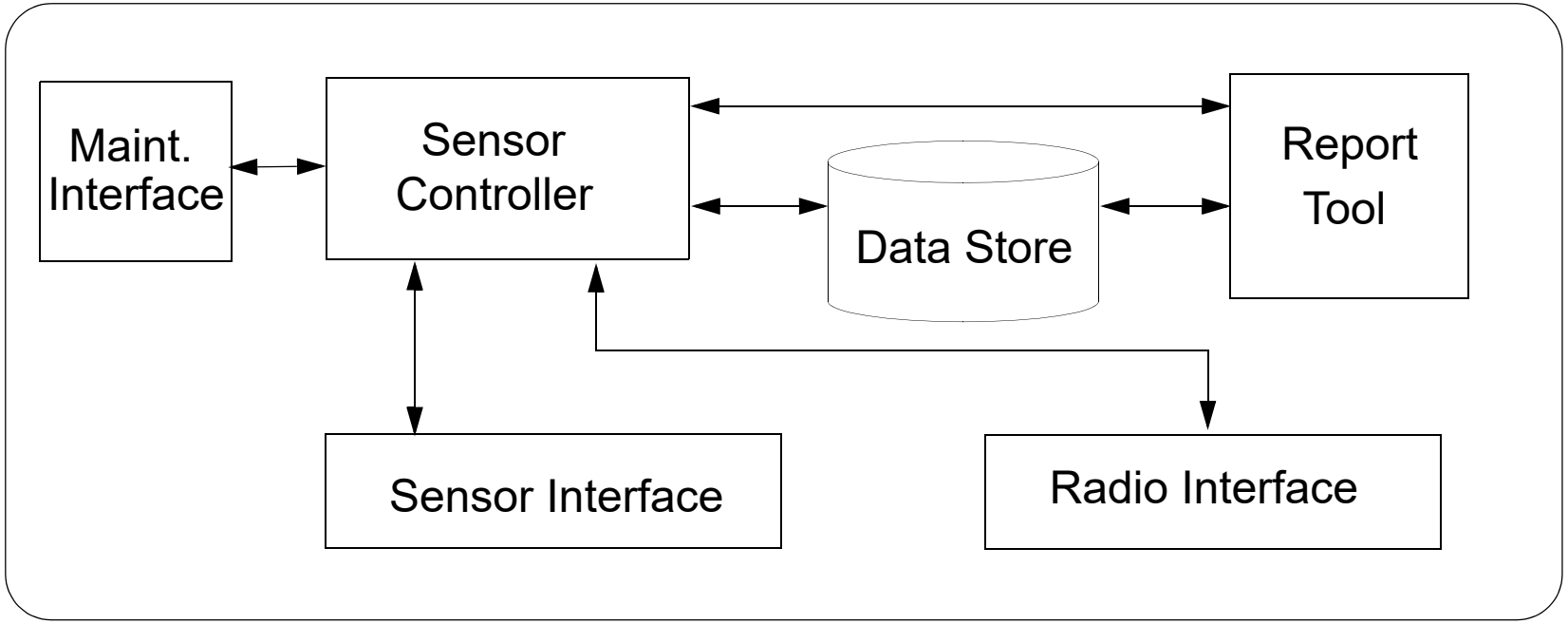


1. commands to hose
2. signals from hose
3. commands to SI
4. response from SI
5. sensor data
6. parameters
7. messages from RI
8. calls to RI
9. signals from radio
10. msgs. from radio
11. raw sensor data
12. formatted data
13. formatted data

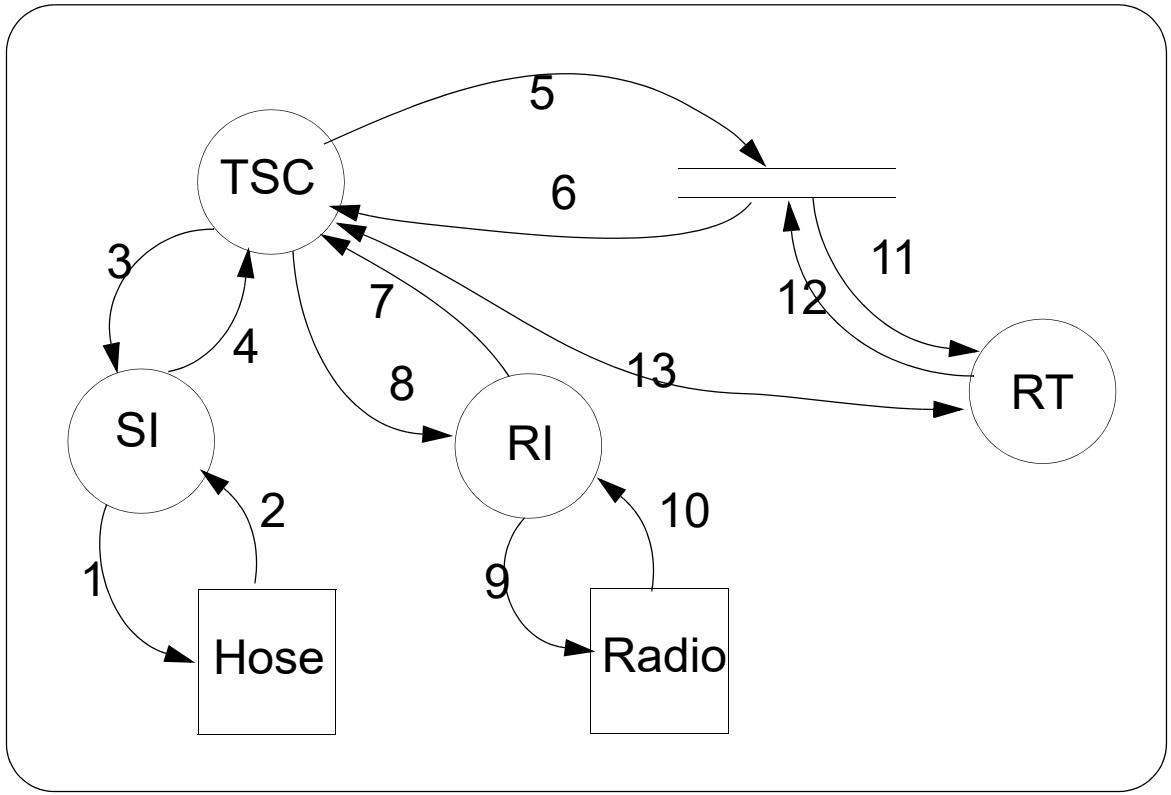
TRAFFIC SENSOR STATES



ALTERNATE SOFTWARE ARCHITECTURE



ALTERNATE TRAFFIC SENSOR DATA FLOW



1. commands to hose
2. signals from hose
3. commands to SI
4. response from SI
5. sensor data
6. parameters
7. messages from RI
8. calls to RI
9. signals from radio
10. msgs. from radio
11. raw sensor data
12. formatted data
13. commands & responses

THINKING ABOUT DESIGN

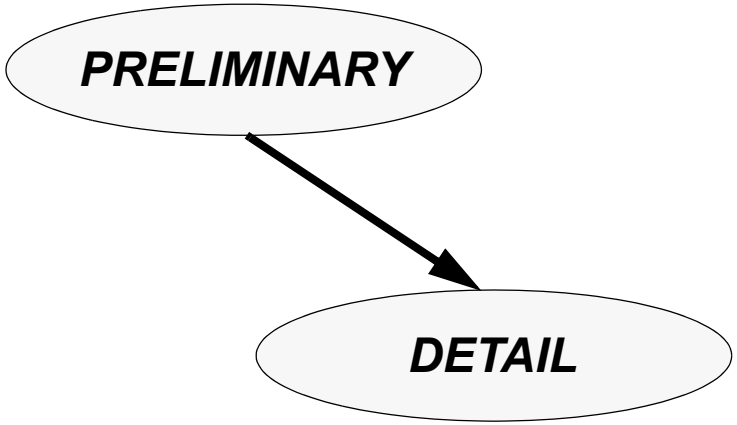
EARLY LIFE CYCLE NEEDS FOR DESIGN

PRODUCT DESIGN: complete, verified specification of hardware and software architecture.

DETAILED DESIGN: complete, verified specification of control structure, data structure, interface relations, key algorithms of each component.

SYSTEM DESIGN TASKS

* ARCHITECTURAL DESIGN
* DATA DESIGN
* PROCEDURAL DESIGN
* INTERFACE DESIGN



DESIGN QUALITY:
What makes one design "good" and another "bad"?

- Efficient organization
- Modular partitioning
- Data and procedure distinct
- Independent modules
- Interfaces which minimize complexity
- Derived from requirements in repeatable manner

THE ART OF DESIGN: Know the User

- TO DESIGN SOFTWARE FOR USERS, KNOW THE USER
- USERS CHANGE THEIR MINDS
- USERS MAKE MISTAKES
- USERS ARE IMPATIENT
- PREFERENCES ARE TRANSITIVE
- USERS CANNOT ALWAYS SAY WHAT THEY WANT BUT THEY KNOW WHAT THEY WANT ONCE THEY SEE IT

THE ART OF DESIGN: Please the User

* **EFFICACIOUS**: Useful and Powerful
* **CONCEPTUAL INTEGRITY**: Simple and Straightforward
* **GRAMMATICAL**: Learnable
* **WELL MAPPED**: Understandable
* **TRUSTWORTHY**: Dependable
* **ENGAGING**: Fun

THE ART OF DESIGN: Methods

* **INVESTIGATION**: Use all available resources to gather information about the problem, the users, and the technology.
* **TAXONOMY**: Thoroughly name all the users' goals and all aspects of the problem.
* **ORGANIZATION**: Search for program functions and user goals that divide into groups congruent with each other.
* **EPIPHANY**: Use creative thinking to find a unifying vision of a product that incorporates the principles of good design.
* **DRAMATIZATION**: Test or validate your design against practical scenarios based on user goals.
* **ITERATION**: Repeat the methods until discrepancies between user goals and software disappear. Then repeat again until users are happy.

KEY DESIGN GOAL: ELEGANCE

i.e., UNIX pipe

cat <fname> | grep “hello”

Use More Than One Design Method

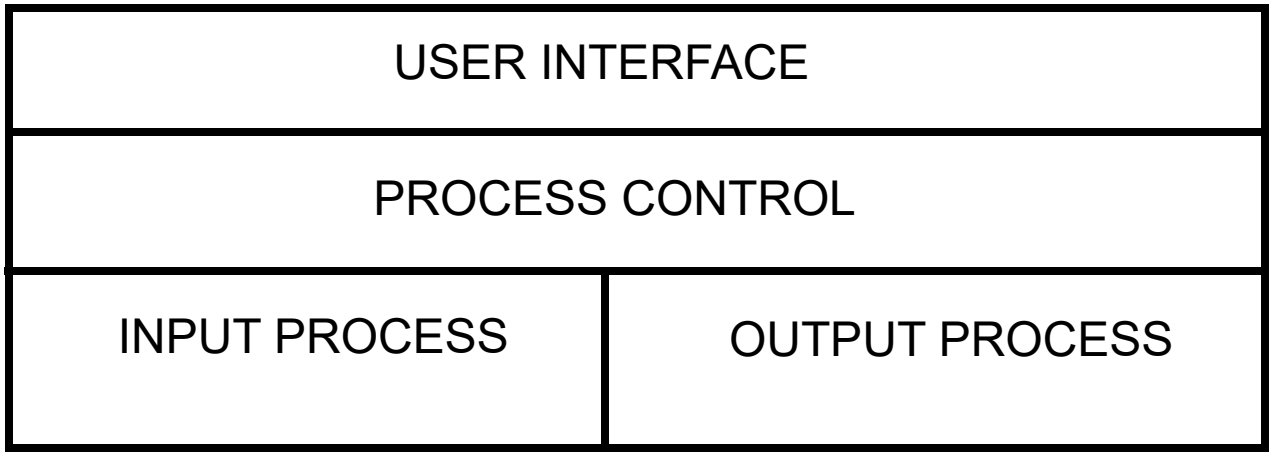
Balance Speed & Rigor

- Reduce interfaces
- Reduce files
- Focus on data content

EXTENDING THE ARCHITECTURE IN DESIGN

* Analysis & requirements provide high level abstraction

* Design steps break system functionality into detailed level



DATA DESIGN ISSUES

* WHAT DATA NEED TO BE CAPTURED?
* WHAT DATA NEED TO BE PRODUCED?
* WHAT ARE THE DATA TRANSFORMS REQUESTED?
* HOW ARE THE DATA MOVED THROUGH THE SYSTEM?

KEY SOFTWARE DESIGN CONCEPTS

* MODULARITY
* SPAN OF CONTROL
* DATA STRUCTURE
* INFORMATION HIDING
* COHESION
* COUPLING

PROCEDURE AND INTERFACE DESIGN

* WITHIN THE ARCHITECTURE HOW WILL DATA BE MANIPULATED

* WHAT ARE THE LOGICAL ALGORITHMS NEEDED

* WHAT ARE ALL THE INTERFACES REQUIRED

  - Presentation

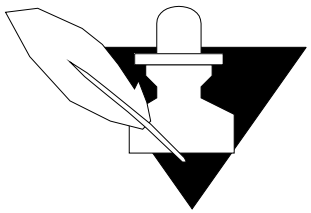
  - Records

  - Communications

STRUCTURED ANALYSIS & STRUCTURED DESIGN

"We should recognize the closed subroutine as one of the greatest software inventions; it has survived three generations of computers and it will survive a few more because it caters to the implementation of one of our basic patterns of abstraction."

Dijkstra, 1972



STRUCTURED DESIGN TECHNIQUES

STRUCTURED DESIGN: KEY TOOLS

- E-R DIAGRAM
- DFD
- STD
- STRUCTURE CHART
- DATA DICTIONARY
- SYSTEM FLOW CHARTS

STRUCTURED DESIGN:
Moving from analysis to solution

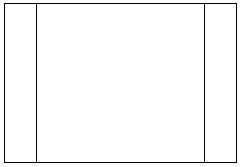
* REDUCING COMPLEXITY BY PARTITIONING GRAPHICALLY

* USE OF SYSTEM MODELING

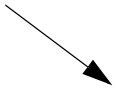
* STRATEGIES TO CONVERT DEFINED PROBLEM TO SOFTWARE

STRUCTURED DESIGN: Notational Methods

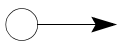
NEW MODULE



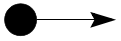
EXISTING MODULE



CALLING RELATIONSHIP



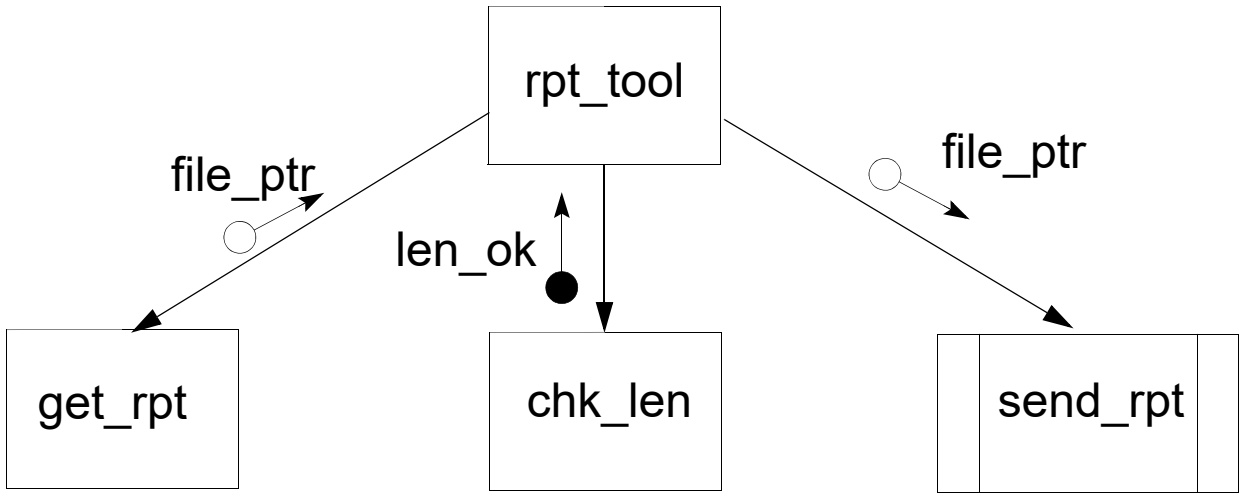
DATA COUPLE

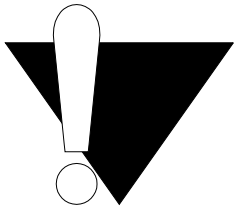


FLAG

TRAFFIC SENSOR EXAMPLE

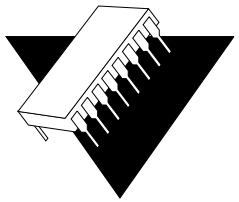
REPORT TOOL STRUCTURE CHART





ANALYZE THIS

PROBLEM STATEMENT:

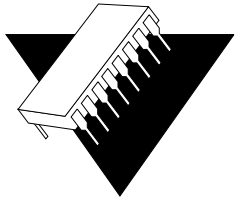


BUILD A DIGITAL BABY

ESSENTIAL MODEL:

- Should respond just like a baby
- Speak nice - it coos
- Shout - it cries
- Goals - cost/size/etc

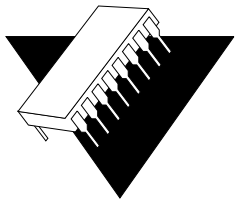
YOU SOLVE IT



BUILD A DIGITAL BABY

* WHAT WILL THE SYSTEM SOLUTION BE?
    - What are the physical components?
    - What abstractions need to be modeled?
    - What will the design be?

AN EARLIER SOLUTION



BUILD A DIGITAL BABY

Naoko Tosa, Musashino Art University, Tokyo

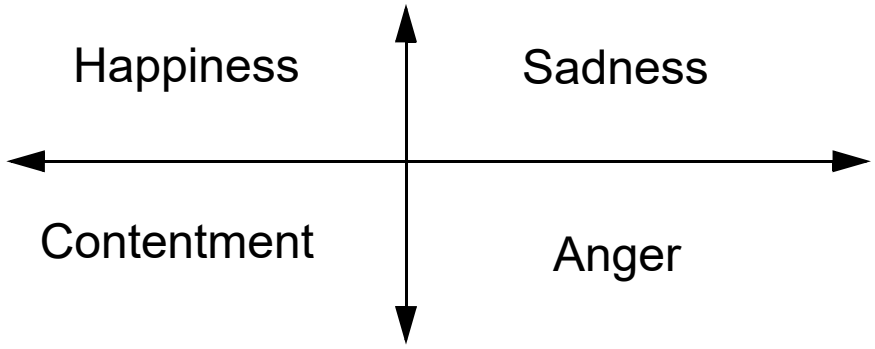
NEURO BABY

- Voice Analyzer
- Image Expression synthesizer
- Voice & Sound Generator

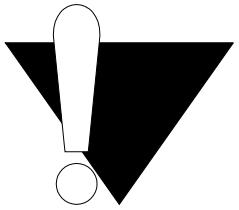
(MultiMedia PC, Neural Net Emulator, Analog-digital encoder)

SOFTWARE SOLUTION FOR NEURO BABY:

1. Teach neural network to recognize inflections to emotions mapping
2. Select emotion variable on emotional plane

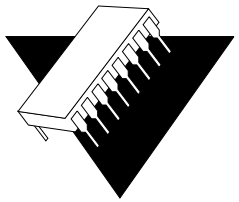


3. Map facial expression & sound to quadrant
4. Display facial expression & play sound



ANALYZE THIS

PROBLEM STATEMENT:

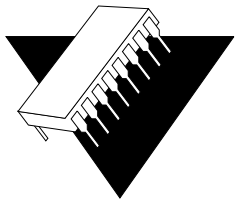


BUILD A GARAGE DOOR OPENER

ESSENTIAL MODEL:

- Open and Close Garage Door Remotely
- Safe/Fast
- Cheap/Reliable

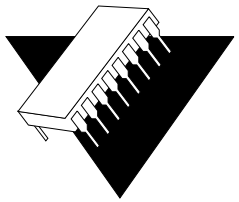
YOU SOLVE IT

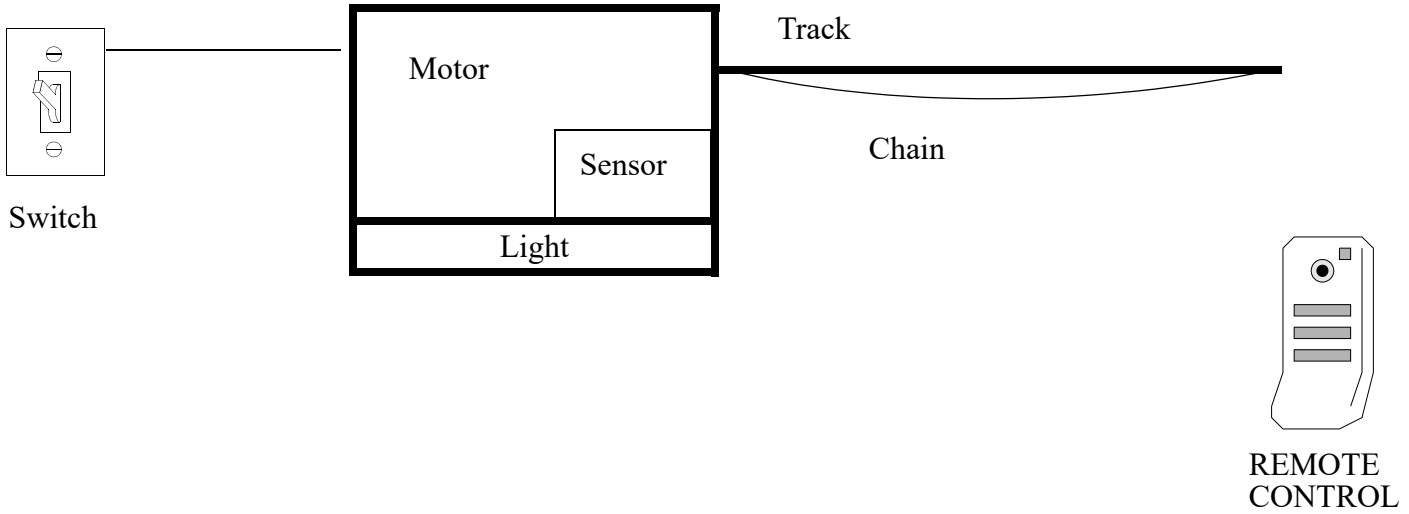


BUILD A GARAGE DOOR OPENER

- Physical Components?
- Models?
- Design?

AN EARLIER SOLUTION





STATE DIAGRAM FOR DOOR OPENER