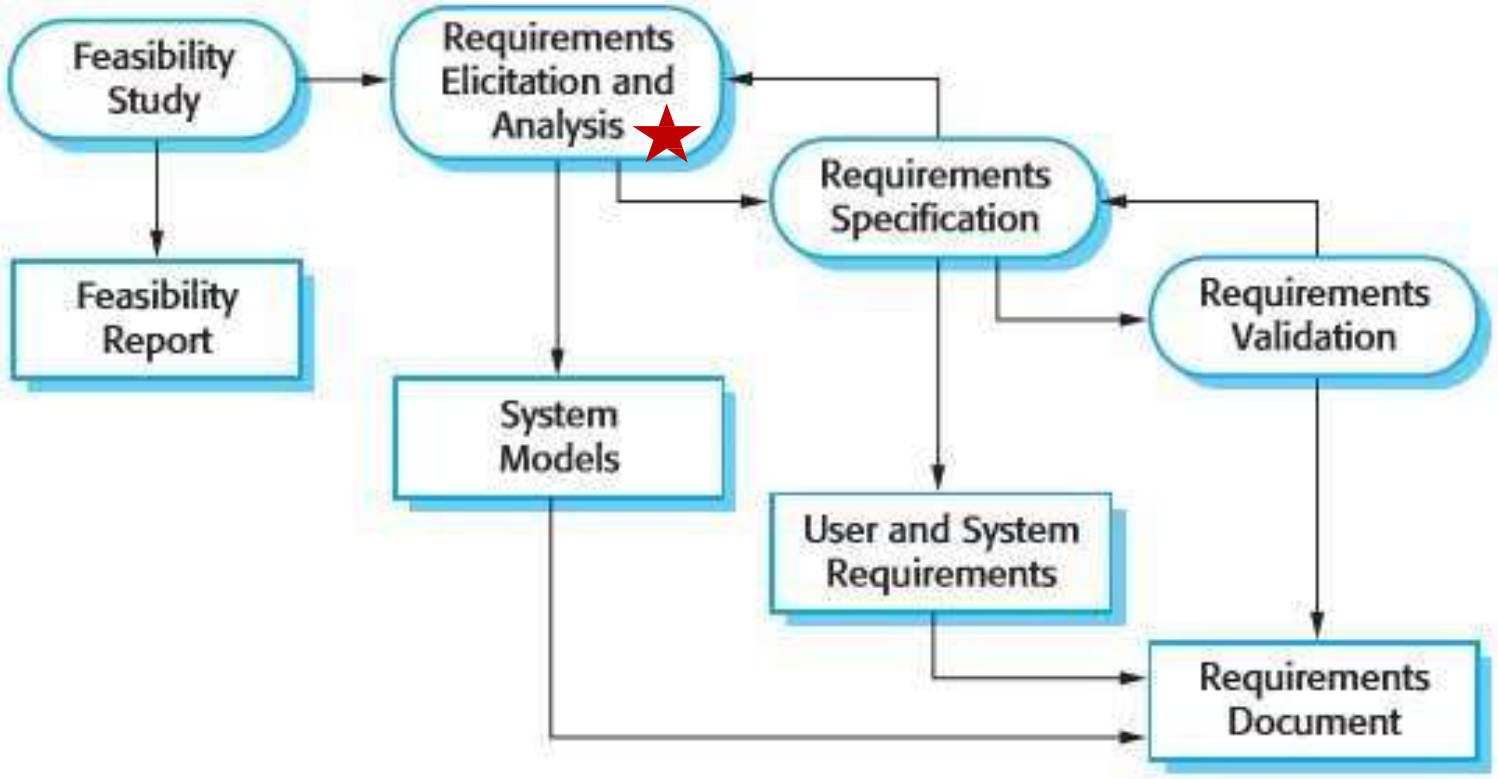
5. Requirements Analysis

Requirements Engineering Process

- Requirements analysis through requirements models
- Requirements prioritization
- Requirements selection (Triage)



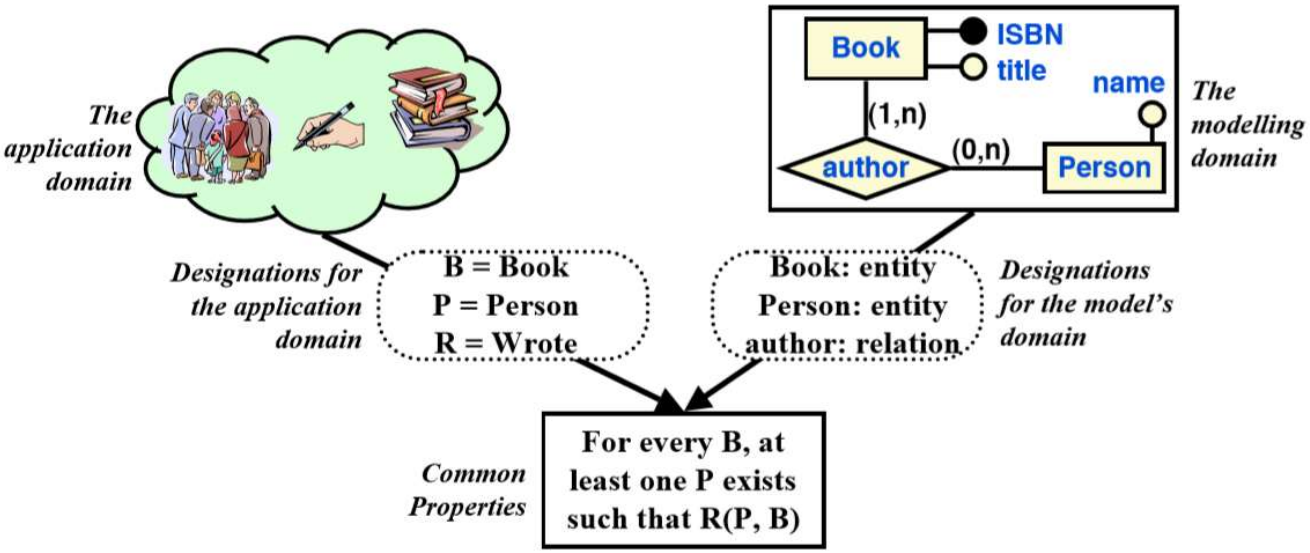
Requirements Modeling

- RE uses **requirement models** to understand the requirements well.
  - Help stakeholders to understand the requirements

- Requirements modelling can
  - Guide **elicitation**
    - It can help you figure out what questions to ask
    - It can help to surface hidden requirements
  - Provide **a measure of progress**
    - Completeness of the models → completeness of the elicitation (might be)
  - Help to **uncover problems**
    - Inconsistency in the models can reveal interesting things
      - E.g., conflicting or infeasible requirements
      - E.g., confusion over terminology, scope, etc.
  - Help us check our **understanding**
    - Reason over the model to understand its consequences
    - Animate the model to help us visualize/validate the requirements

Correspondence of Requirements Models

- The requirement model is only useful, when
  - The **model**'s phenomena correspond in a systematic way to the phenomena of the **domain** being modelled.
  - There will always be:
    - ***Phenomena in the model*** *that are not present in the application domain*
    - ***Phenomena in the application domain*** *that are not in the model*
  - All models are **inaccurate** (to some extent).



Features of Requirements Models

- **Features of good requirements models**
  - **Complete**
    - Modeling guides elicitation
    - Completeness of the model leads to completeness of elicitation
  - **Consistency**
    - Modeling uncovers problems
    - Inconsistency in modeling implies omission, conflict, disagreement and ambiguity
  - **Testability**
    - Modeling checks for expected qualities and predicts end result

- Principles of requirements modeling
  - Early artifacts
  - Cheap to make
  - Easy to visualize and optimize

Requirements Modelling Notations

- **Natural language**
  - Extremely expressive and flexible
    - Useful for elicitation, and to annotate models for readability
  - Poor at capturing key relationships

- **Semi-formal notation**
  - Captures structure and some semantics
  - Can perform (some) reasoning, consistency checking, animation, etc.
    - E.g., diagrams, tables, structured English, etc.
    - E.g., **UML**

- **Formal notation**
  - Precise semantics, extensive reasoning possible
    - Underlying mathematical model (e.g., set theory, FSMs, CSP, etc.)
  - Very detailed models
    - May be more detailed than we need
    - RE formalisms are for conceptual modelling, hence differ from most computer science formalisms

Principles for Modelling Notations

Implementation Independence	• Does not model data representation, internal organization, etc.
Abstraction	• Extracts essential aspects
Formality	• Unambiguous syntax • Rich semantic theory
Constructability	• Can construct pieces of the models to handle complexity and size
Ease of analysis	• Ability to analyze for ambiguity, incompleteness and inconsistency
Traceability	• Ability to cross-reference elements • Ability to link to design, implementation, etc.
Executability	• Can animate (visualize) the model, to compare it to reality
Minimality	• No redundancy of concepts in the modelling scheme

A Survey of Modelling Techniques

- **Modelling Enterprises**
  - Goals & objectives
  - Organizational structure
  - Tasks & dependencies
  - Agents, roles, intentionality

Organization modelling:
i*, SSM, ISAC
Goal modelling:
KAOS, CREWS

- **Modelling Information & Behavior**
  - Information Structure
  - Behavioral views
    - Scenarios and Use Cases
    - State machine models
    - Information flow
  - Timing/Sequencing requirements

Information modelling:
E-R, Class Diagrams
Structured Analysis:
SADT, SSADM, JSD → SASD
Object Oriented Analysis:
OOA, OOSE, OMT → OOAD & UML
Formal Methods:
SCR, NuSCR, Statecharts, Z, Larch, VDM…

- **Modelling System Qualities (NFRs)**
  - All the 'ilities':
    - usability, reliability, evolvability, safety, security, performance, interoperability,

Quality tradeoffs:
QFD, win-win, AHP
Specific NFRs:
Timed Petri nets (performance)
Task models (usability)
Probabilistic MTTF (reliability)

The State-of-the-Art Requirements Modeling Methods

1. Structured analysis

- Data Flow Diagram (**DFD**)
- State Transition Diagram (**STD**)
- Entity-Relation Diagram (**ERD**)

2. Use Case analysis

- Use Case Modeling (**UC**)

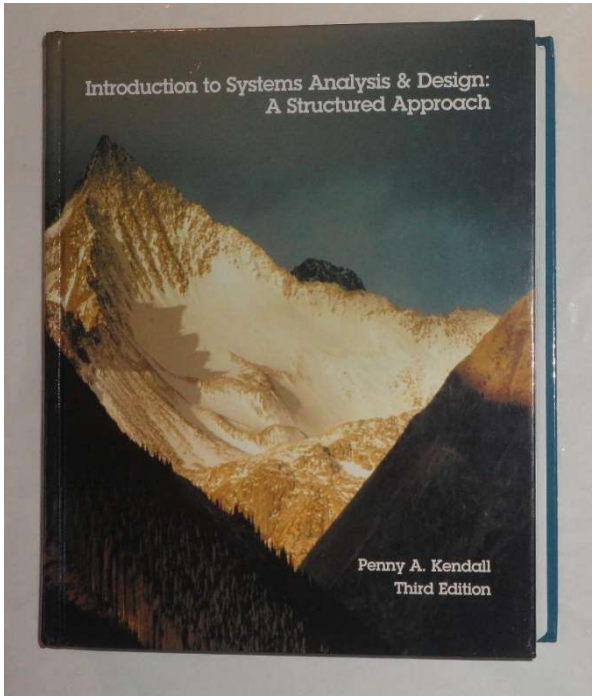
3. Goal and Scenario based analysis

- Goal-Scenario Modeling (**GS**)

1. Structured Analysis

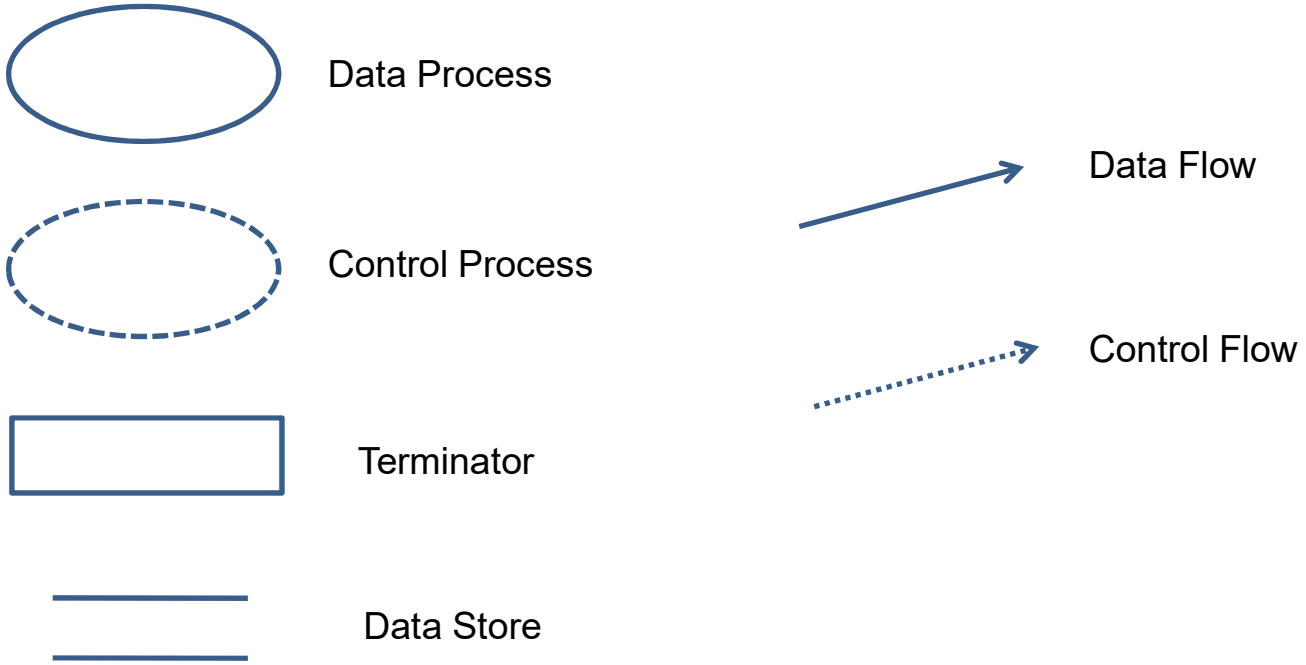
- **Structured analysis** [Kendall 1996]
  - A set of **techniques** and **graphical** tools
    - Allowing the analysts to develop a new kind of system specification that are easily understandable to the users.
  - Data/Functional modeling: **DFD**, **ERD**
  - State-oriented modeling: **STD**

- Analysts attempt to divide large, complex problems into smaller, more easily handled ones.
  - **Top-Down Divide and Conquer** approach



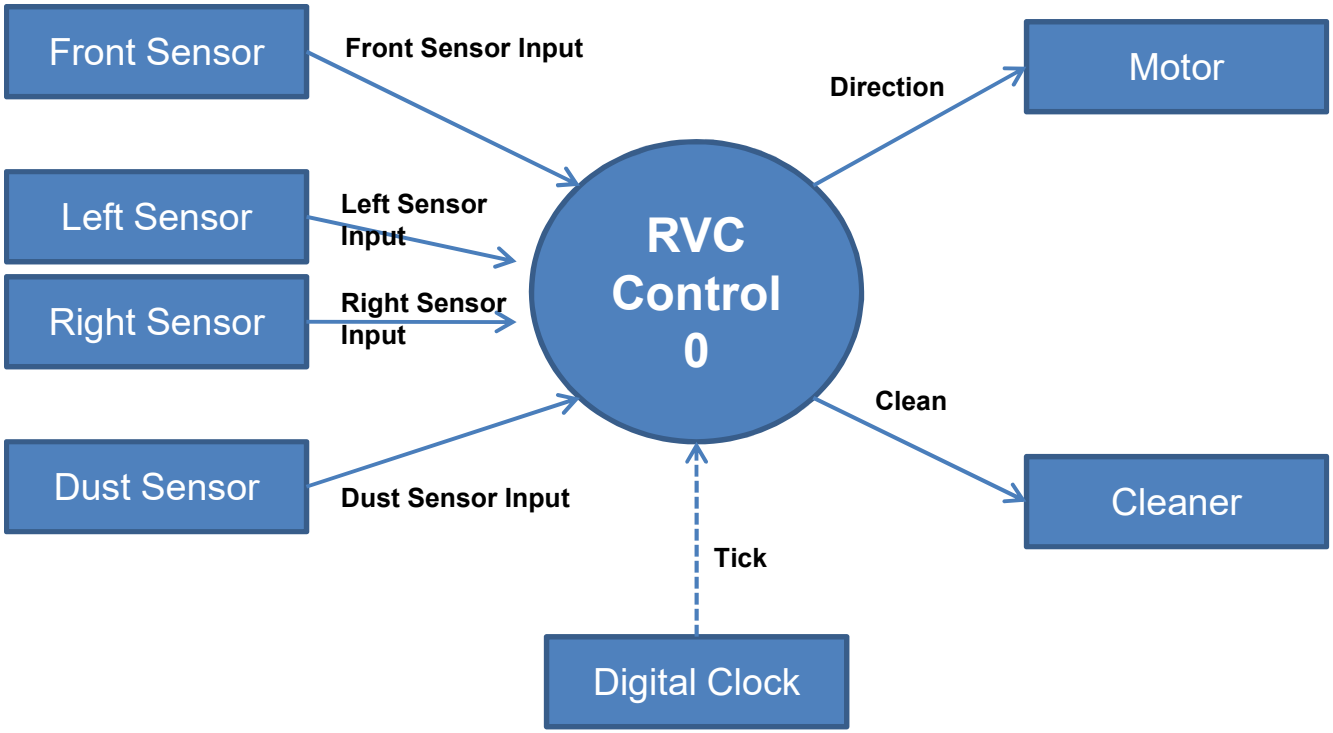
Data Flow Diagram (DFD)

- Provides a means for **functional decomposition**
  - Composed of hierarchies (levels) of DFDs
- Model Elements



DFD Level 0 - RVC Example

- System context diagram

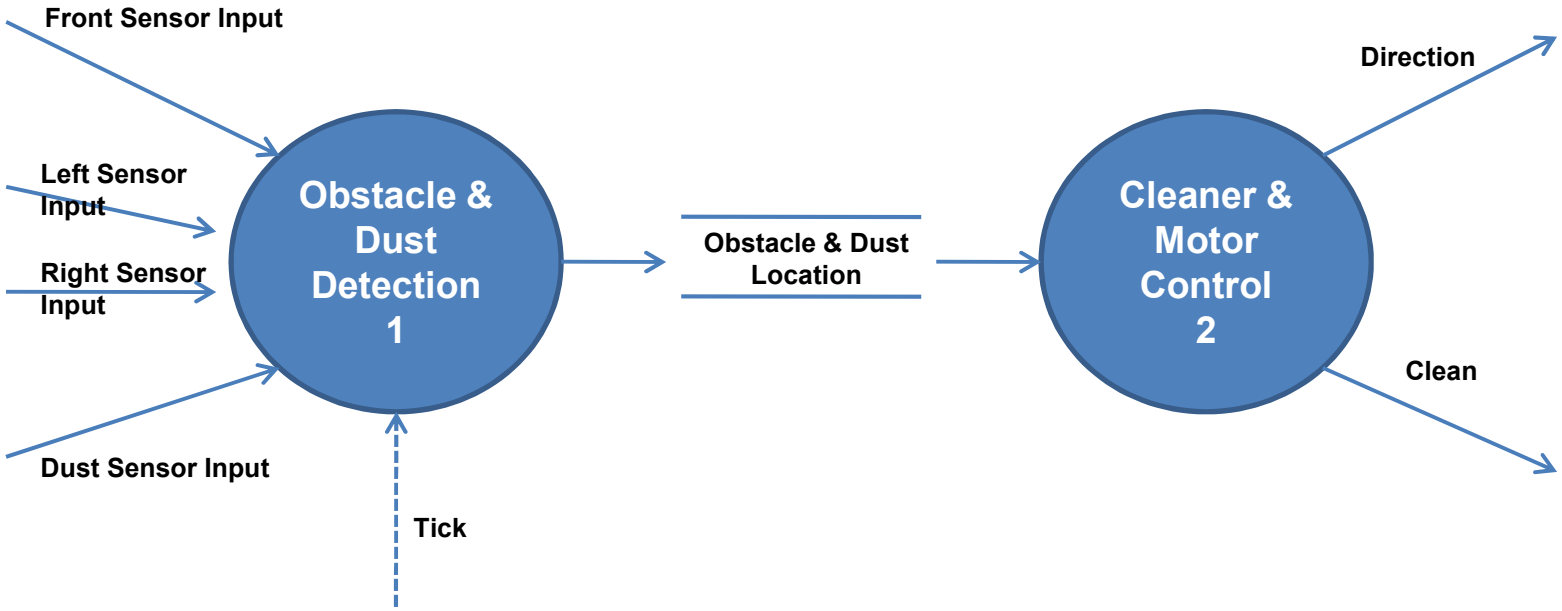


DFD Level 0 - RVC Example

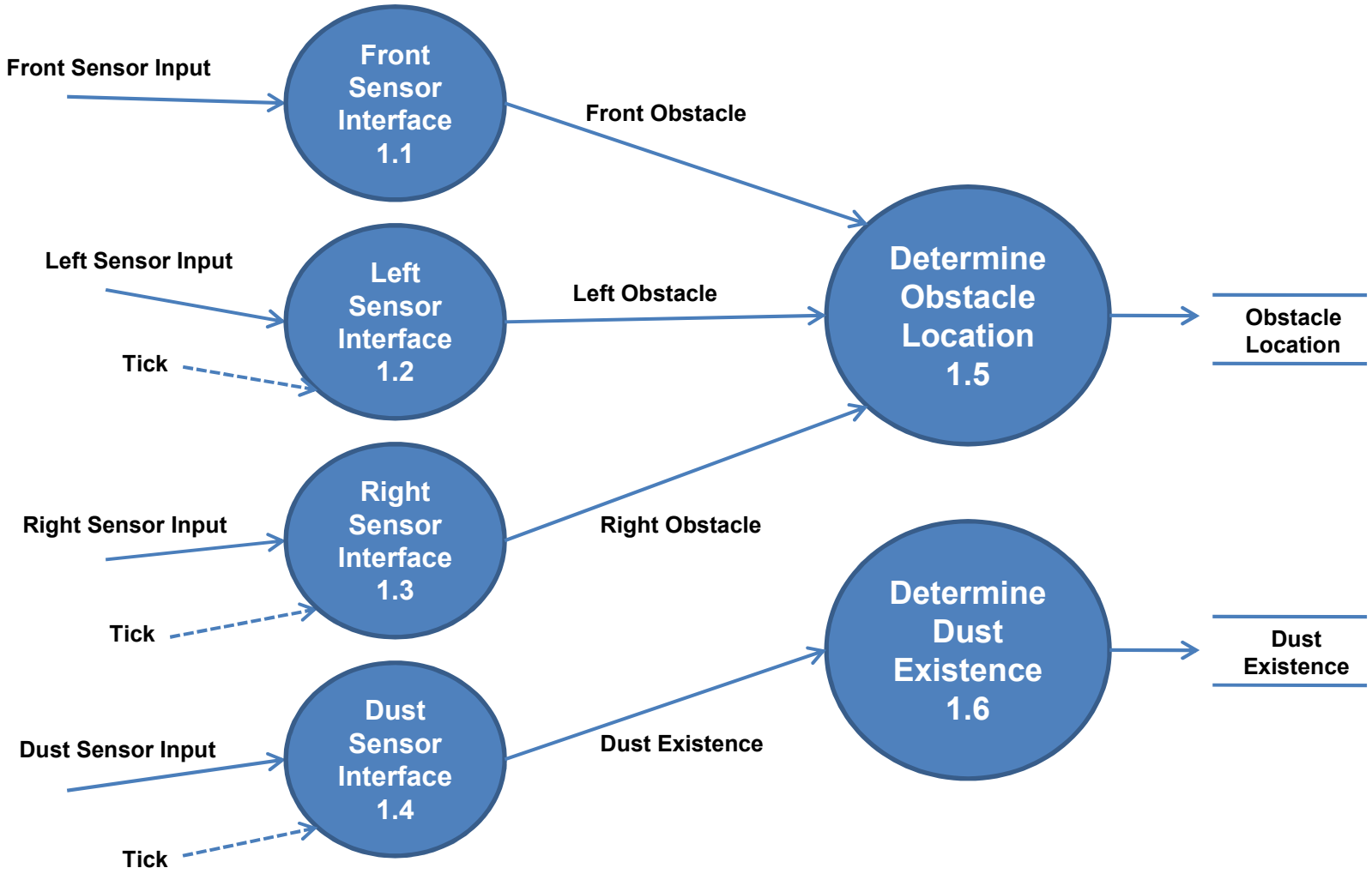
- (A kind of) Data Dictionary

Input/ Output Event	Description	Format / Type
Front Sensor Input	Detects obstacles in front of the RVC	True / False , Interrupt
Left Sensor Input	Detects obstacles in the left side of the RVC periodically	True / False , Periodic
Right Sensor Input	Detects obstacles in the right side of the RVC periodically	True / False , Periodic
Dust Sensor Input	Detects dust on the floor periodically	True / False , Periodic
Direction	Direction commands to the motor (go forward / turn left with an angle / turn right with an angle)	Forward / Left / Right / Stop
Clean	Turn off / Turn on / Power-Up	On / Off / Up

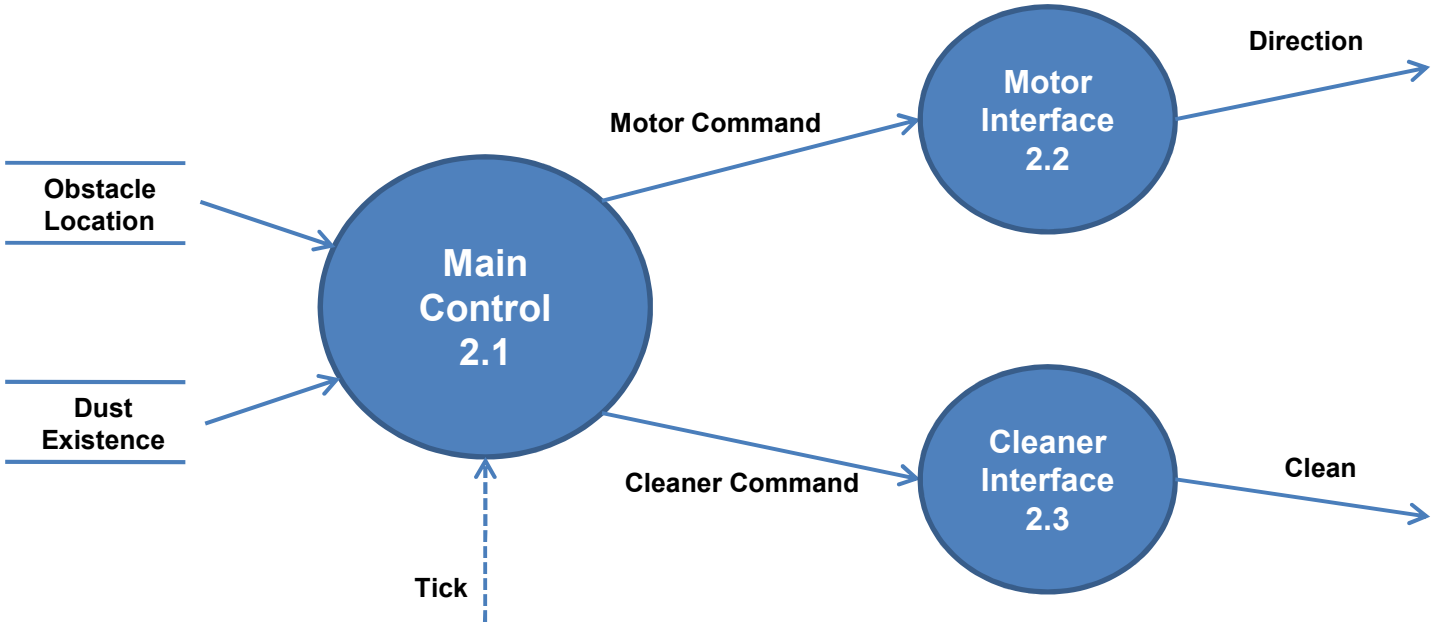
DFD Level 1 - RVC Example



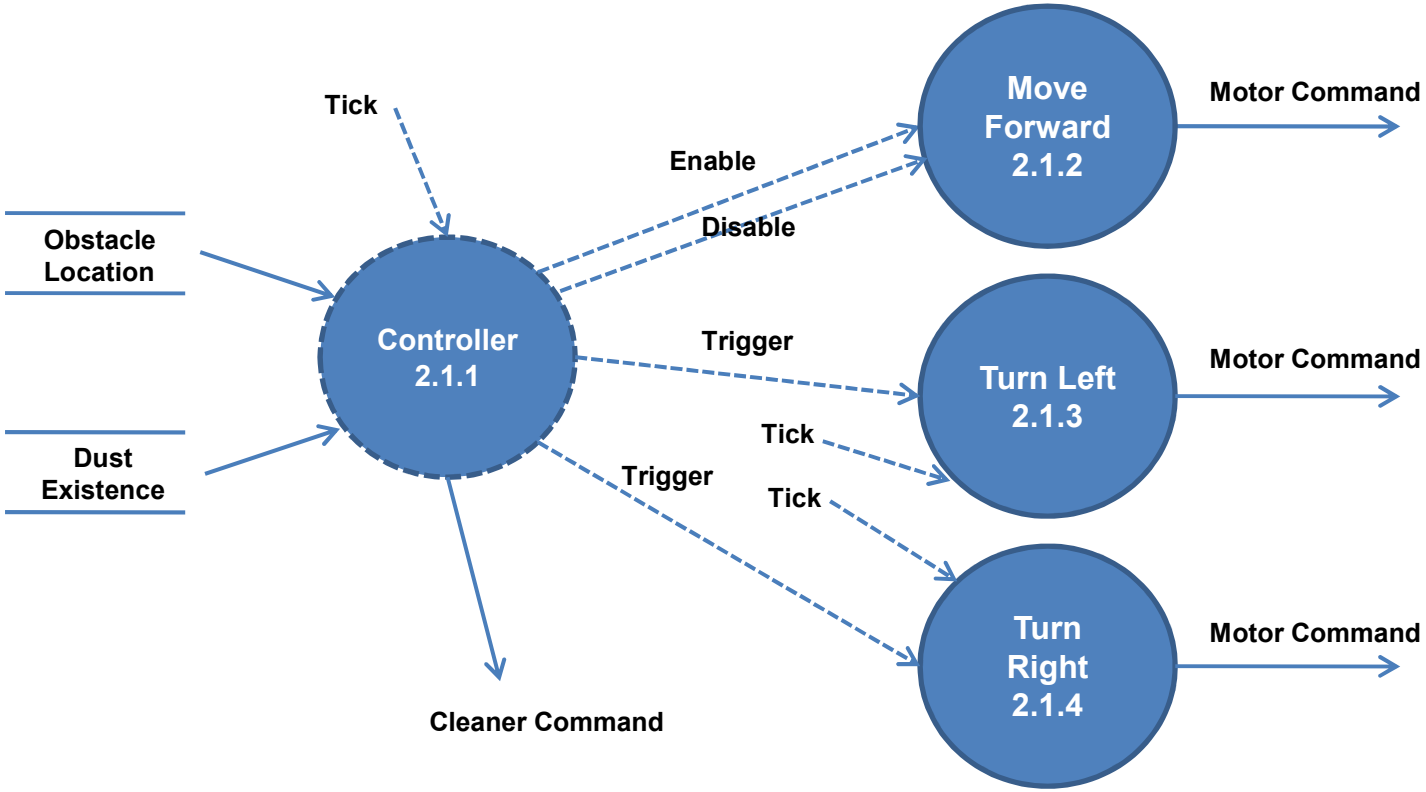
DFD Level 2 - RVC Example



DFD Level 2 - RVC Example

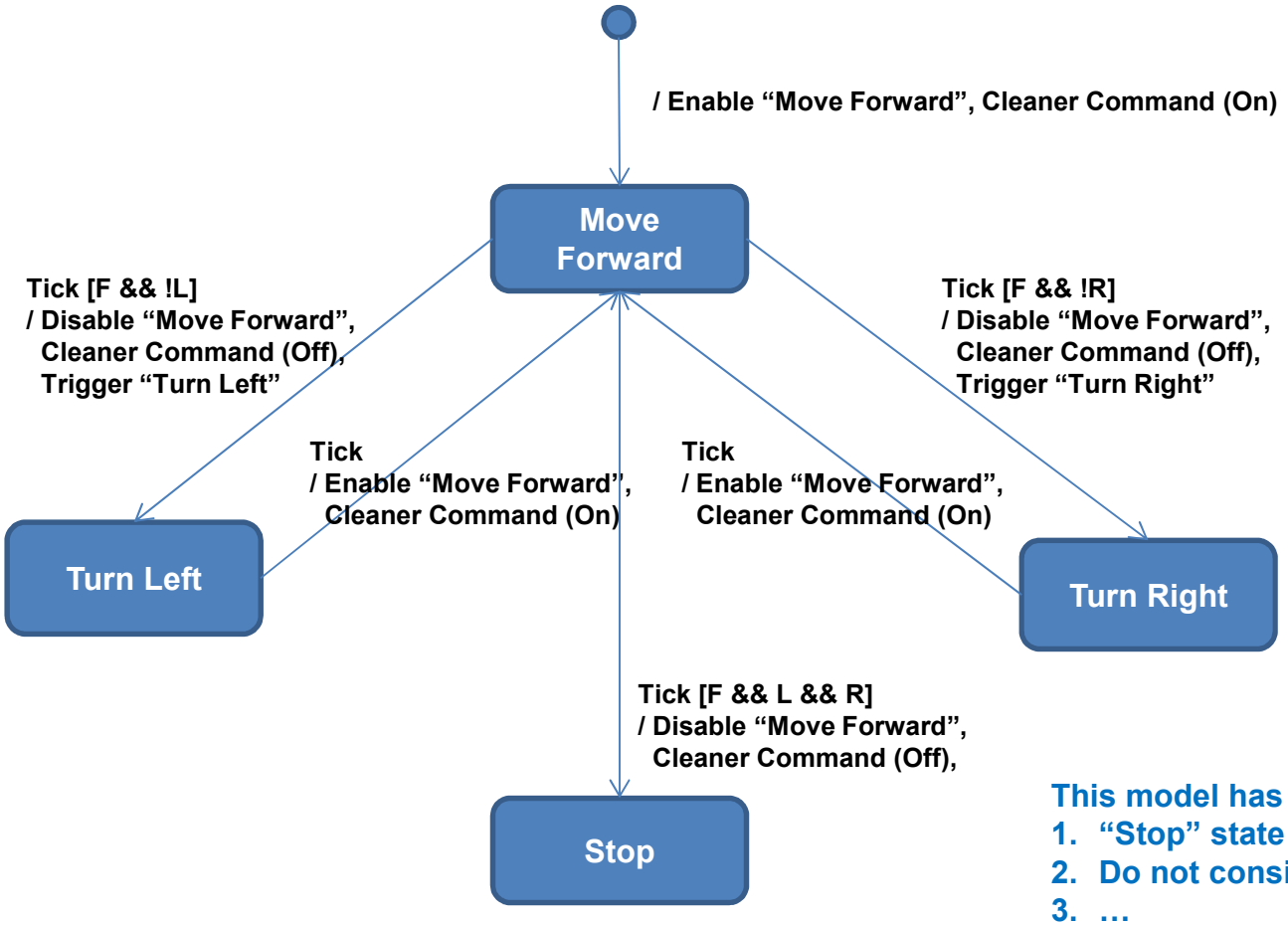


DFD Level 3 - RVC Example



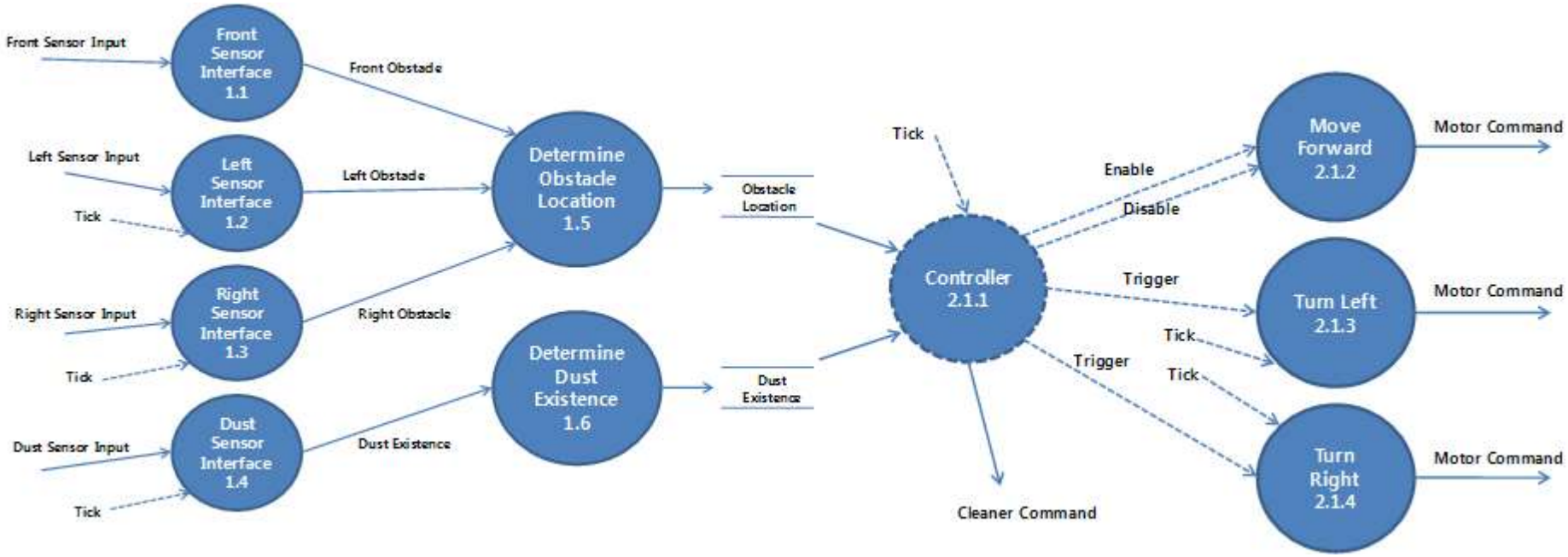
DFD Level 4 - RVC Example

- STD (State Transition Diagram) for Controller 2.1.1



This model has many seeded problems
1. “Stop” state is not normal
2. Do not consider “Dust”
3. …

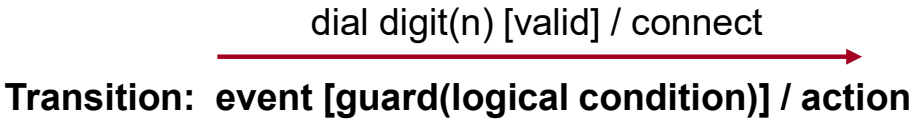
DFD - RVC Example



State Transition Diagram (STD)

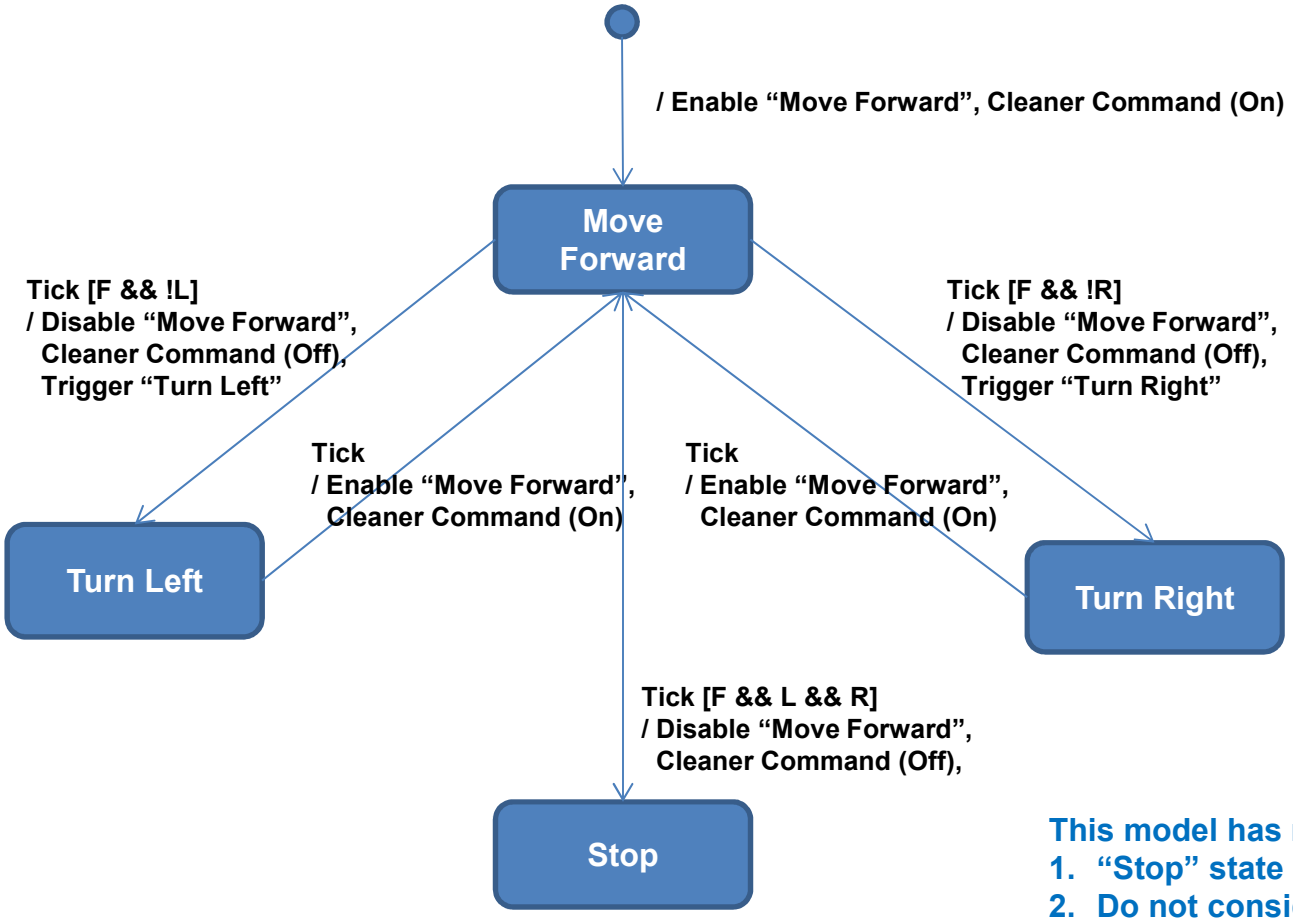
- **State Transition diagram** is used
  - to model the possible states of a system or controller
  - to show how state transitions occur as a consequence of events
  - to show what behavior the system exhibits in each state

- Model Elements



State: State Name

- STD (State Transition Diagram) for RVC Controller 2.1.1



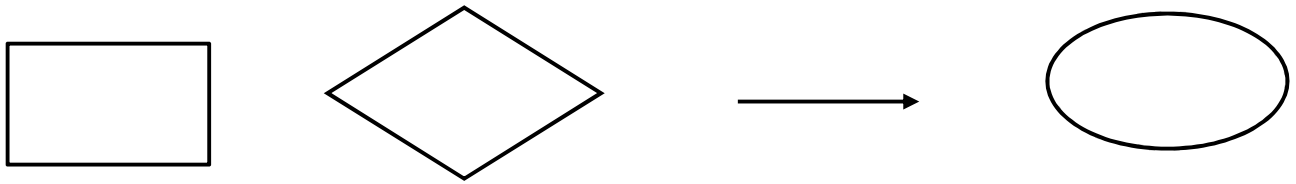
This model has many seeded problems

1. **“Stop” state is not normal**
2. **Do not consider “Dust”**
3. **…**

E-R Modeling

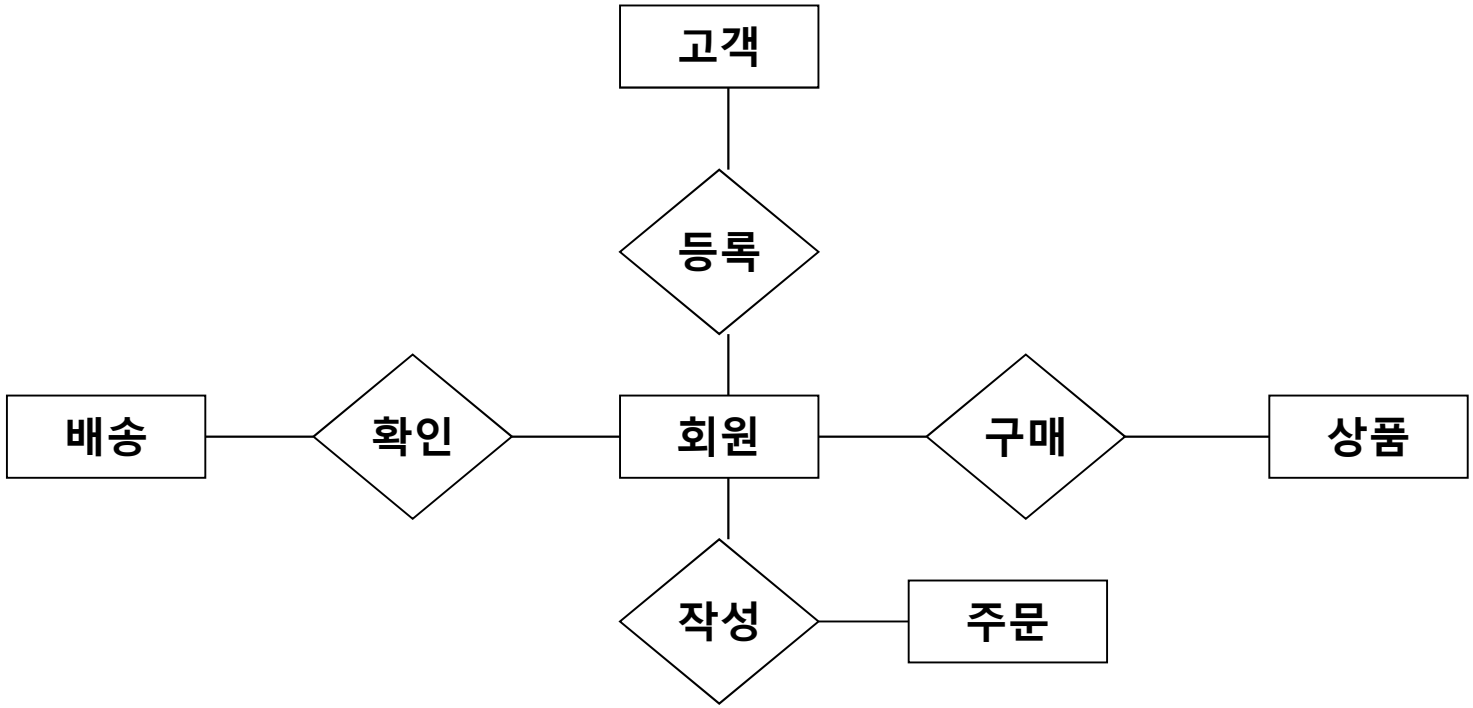
- A graphical representation of the **data layout** of a system at a high level of abstraction
  - Defines data elements and their inter-relationships in the system.
  - Similar with the class diagram in UML.

- Model Elements



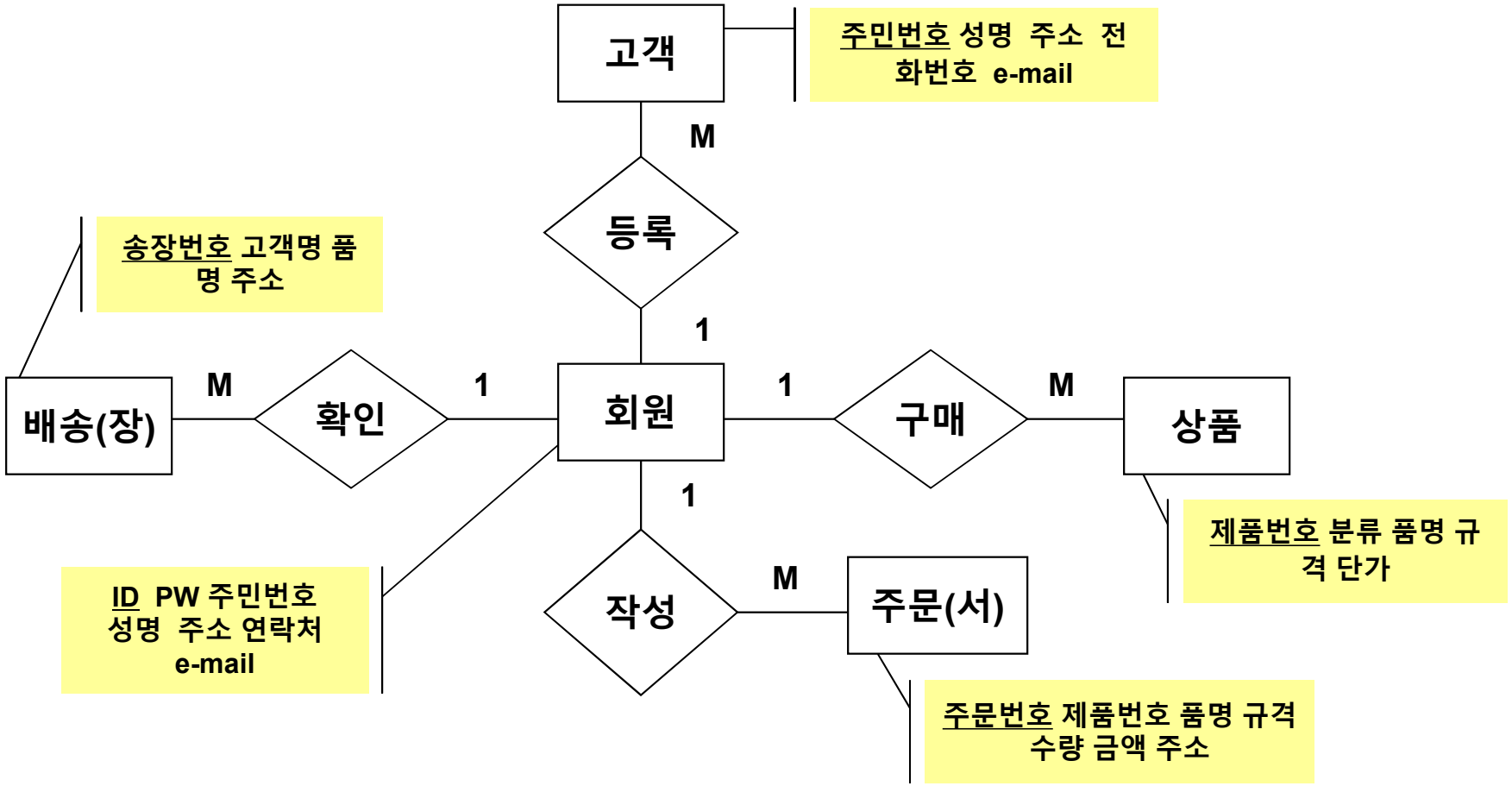
E-R Modeling

- The 1st version : Shopping process at Malls



E-R Modeling

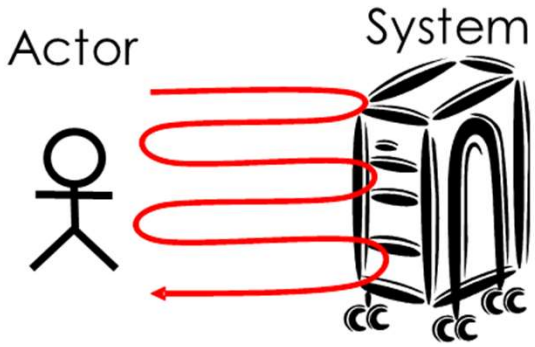
- The 2nd version



2. Use Case Analysis

- **Use cases** are **text stories** of some actors using a system to meet goals.
  - A mechanism to capture (analyzes) requirements
  - An example:
    - **Process Sale**: A customer arrives at a checkout with items to purchase. The cashier uses the POS system to record each purchased item. The system presents a running total and line-item details. The customer enters payment information, which the system validates and records. The system updates inventory. The customer receives a receipt from the system and then leaves with the items.
  - Use case is not a diagram, but a text.

- **Use case diagram** illustrates the name of use cases and actors, and the relationships between them.
  - System context diagram
  - A summary of all use cases



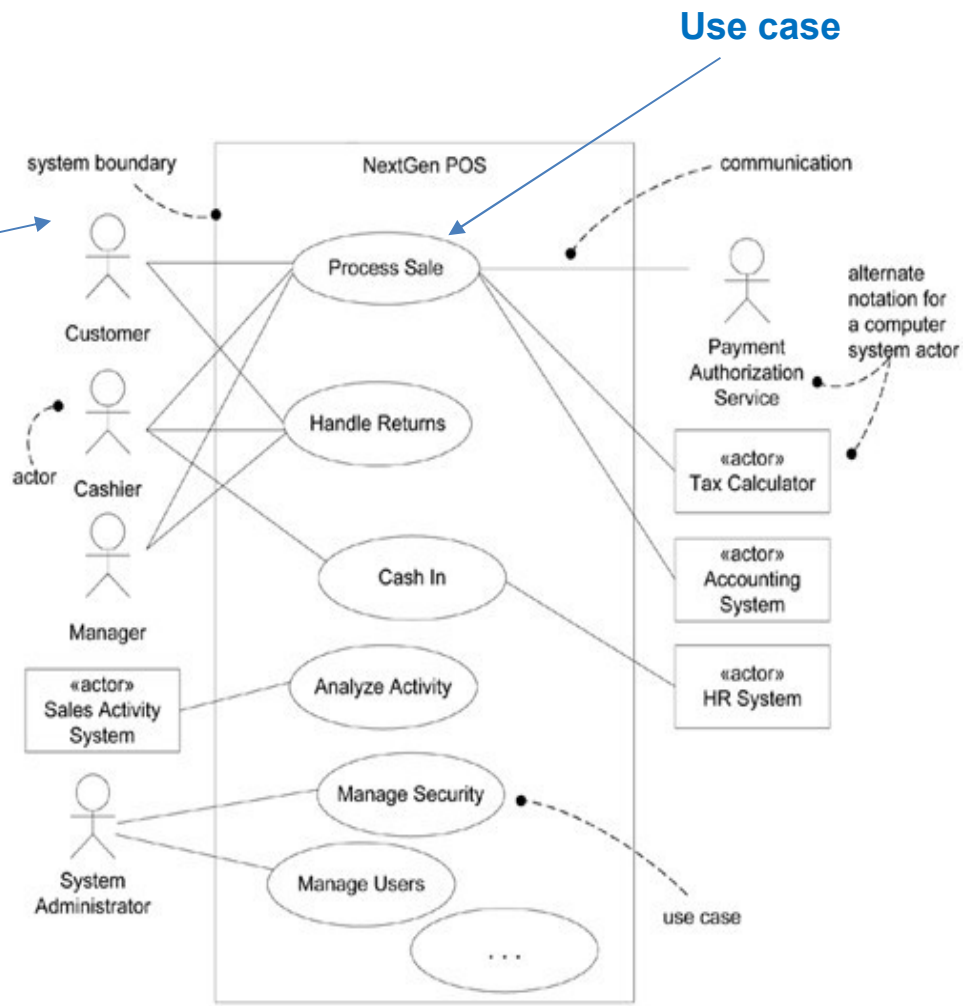
Actor

Something with behavior, such as a person, computer system, or organization

- Primary Actor : has user goals fulfilled through using services of the SuD (System Under Discussion) , *e.g.,* cashier

- Supporting Actor : provides a service to the SuD, *e.g.,* payment authorization service

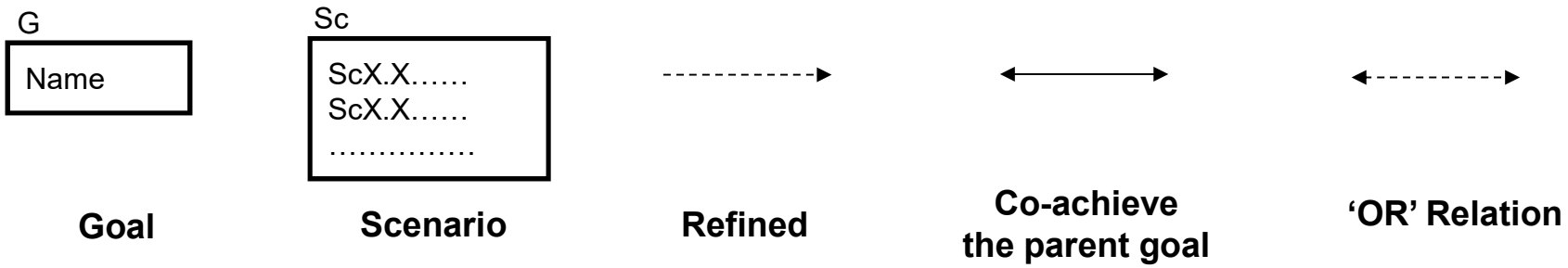
- Offstage Actor : has an interest in the behavior of the use case, but is not primary or supporting, *e.g.,* tax agency



3. Goal-Scenario Based Analysis

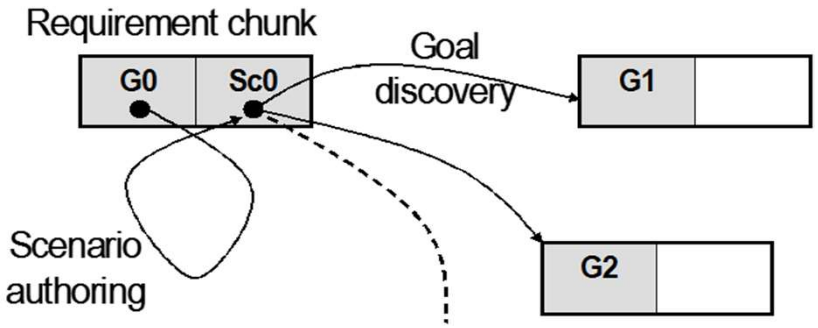
- An analysis using **goal** and **scenario models** to express and refine requirements
  - Provides rationale for the requirements
  - Supports requirements analysis through scenarios
    - Storyline and example based description
  - Refines goals through scenarios

- Model Elements



Goal & Scenario

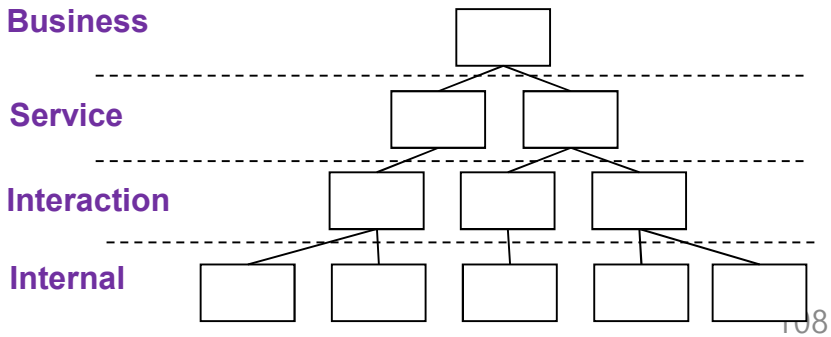
- **Goal**
  - High-level abstraction requirements
  - Example: “유비쿼터스 기술이 접목된 ATM 서비스를 제공한다.”
- **Scenario**
  - Purposeful interaction between entities
  - Example: “사용자는 ATM으로부터 현금을 인출한다.”
- The relationship between goal and scenario
  - Goal are achieved by scenarios
  - Goals are explained by scenarios
    - Goals are abstract
    - Scenarios are concrete



Goal & Scenario Modeling

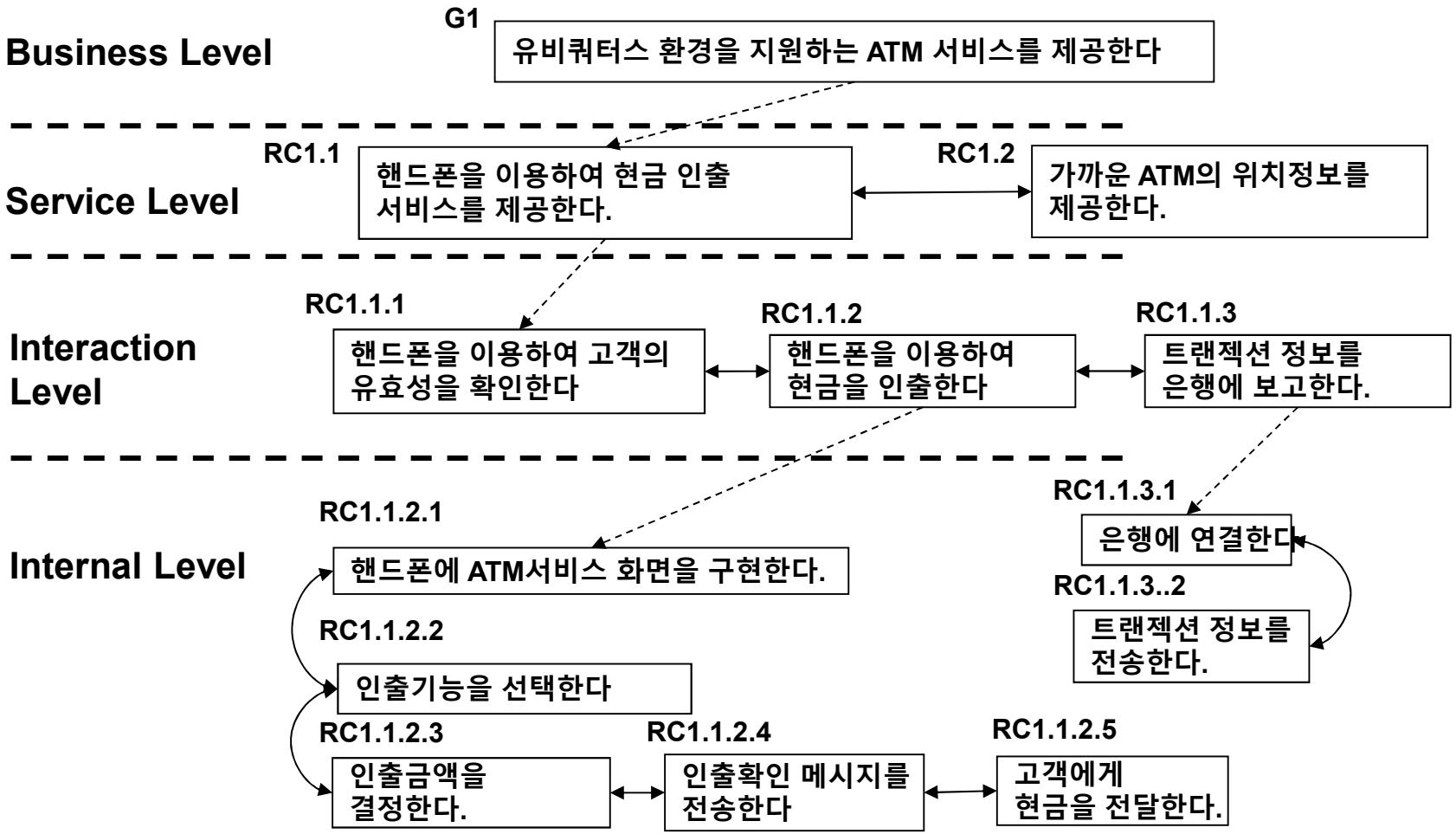
- Inputs: Initial requirements
- Outputs: **Goal tree**
  - Abstraction levels provide separation of concern and levels of goal & scenario modeling

- The 4 abstraction levels
  - **Business** : Represents the ultimate purpose of a system
  - **Service** : Represents the services that a system should provide to an organization and the rationale
  - **Interaction** : Represents the interaction between system and external agent (user or external system)
  - **Internal** : Represents what the system needs to perform the interactions selected at the user level



Example of G&S Modeling

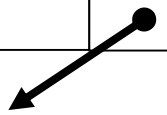
- A partial example of an ATM system



RC1.1.2

(핸드폰을 이용하여)$_{way}$ (ATM으로부터)$_{direction}$
(현금을)$_{target}$ (인출한다)$_{action}$

SC1.1



1. (ATM이)$_{subject}$ (고객의 핸드폰에)$_{direction}$ (ATM 서비스 화면을)$_{target}$ (전송한다)$_{action}$
2. (핸드폰은)$_{subject}$ (고객에게)$_{direction}$ (ATM서비스 화면을)$_{target}$ (출력한다)$_{action}$
3. (고객이)$_{subject}$ (인출기능을)$_{target}$ (선택한다)$_{action}$
4. (핸드폰이)$_{subject}$ (고객에게)$_{direction}$ (인출금액 입력화면을)$_{target}$ (출력한다)$_{action}$
5. (고객이)$_{subject}$ (핸드폰에)$_{direction}$(인출금액을)$_{target}$ (입력한다)$_{action}$
   만약 인출금액이 올바르면 (state)
6. (핸드폰은)$_{subject}$ (고객에게)$_{direction}$ (인출확인 메세지를)$_{target}$ (전송한다)$_{action}$
7. (ATM이)$_{subject}$ (고객에게)$_{direction}$ (현금을)$_{target}$ (전달한다.)$_{action}$

Requirements Prioritization

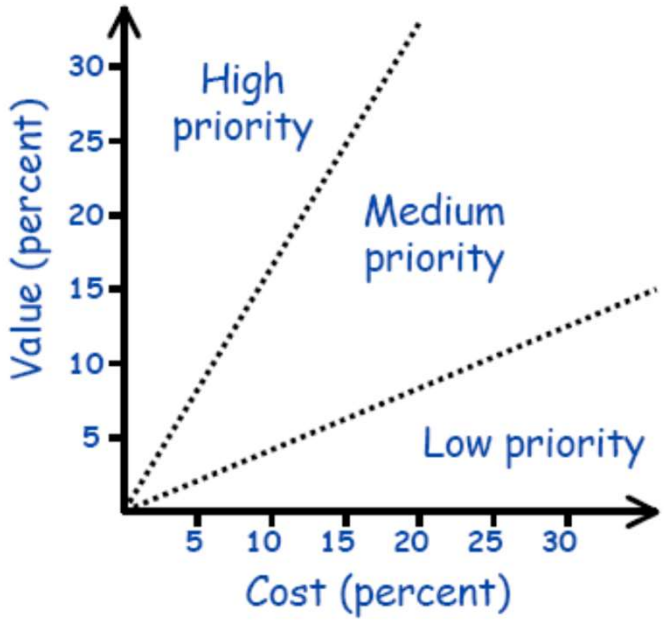
- **Need to select what to implement, after analyzing requirements**
  - Customers (usually) ask for too much
  - Balance time-to-market with amount of functionality
  - Decide which features go into the next release

- For each requirement/feature, ask:
  - How important is this to the customer?
  - How much will it cost to implement?
  - How risky will it be to attempt to build it?

- **Perform Triage:**
  - Some requirements must be included
  - But, some requirements should definitely be excluded

A Cost-Value Approach

- **Calculate return on investment (ROI)**
  1. Assess each requirement's importance (value) to the project as a whole
  2. Assess the relative cost of each requirement
  3. Compute the cost-value trade-off:



Estimating Cost & Value

- Two approaches:
  - **Absolute scale** (e.g., dollar values)
    - Requires much domain experience
  - **Relative values** (e.g., less/more; a little, somewhat, very)
    - Much easier
    - Prioritization becomes a sorting problem
      - Bubble sort
      - Binary sort
      - MST (Minimum Spanning Tree)

Complications on Estimation

- **Hard to quantify differences quantitatively**
  - Easier to say “x is more important than y” than to estimate by how much
- **Not all requirements comparable**
  - E.g., different levels of abstraction
  - E.g., core functionality vs. customer enhancements
- **Requirements may not be independent**
  - No point selecting between X and Y if they are mutually dependent
- **Stakeholders may not be consistent**
  - E.g., if X > Y, and Y > Z, then presumably X > Z ?
- **Stakeholders might not agree**
  - Different cost/value assessments for different types of stakeholder

Analytic Hierarchy Process (AHP)

- **The AHP Process**
  1. **Create n x n matrix for n requirements**
     - For element (x, y) in the matrix enter:
       - 1 : if x and y are of equal value
       - 3 : if x is slightly more preferred than y
       - 5 : if x is strongly more preferred than y
       - 7 : if x is very strongly more preferred than y
       - 9 : if x is extremely more preferred than y
       - Use the intermediate values (2,4,6,8), if compromise needed
     - For (y, x) enter the reciprocal.
  2. **Estimate the eigenvalues:**
     - Use your own approach (strategy, heuristics)
     - E.g., “averaging over normalized columns”
       1. Calculate the sum of each column
       2. Divide each element in the matrix by the sum of it’s column
       3. Calculate the sum of each row
       4. Divide each row sum by the number of rows
     - This gives a value for each requirement:
       - Giving the estimated percentage of total value of the project

AHP Example - Estimating Costs

	Req1	Req2	Req3	Req4
Req1	1	1/3	2	4
Req2	3	1	5	3
Req3	1/2	1/5	1	1/3
Req4	1/4	1/3	3	1

Normalise columns

	Req1	Req2	Req3	Req4
Req1	0.21	0.18	0.18	0.48
Req2	0.63	0.54	0.45	0.36
Req3	0.11	0.11	0.09	0.04
Req4	0.05	0.18	0.27	0.12

Sum the rows

sum	sum/4
1.05	0.26
1.98	0.50
0.34	0.09
0.62	0.16

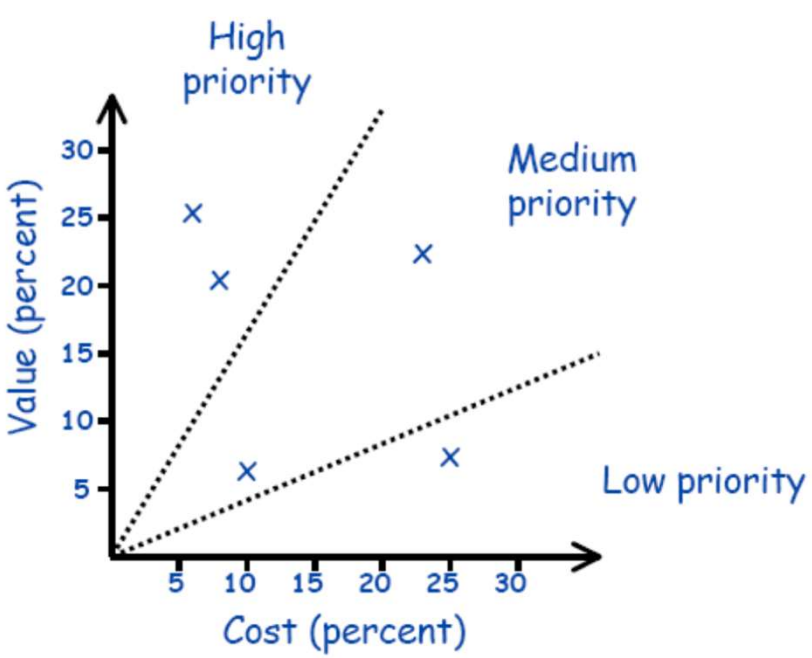
Result

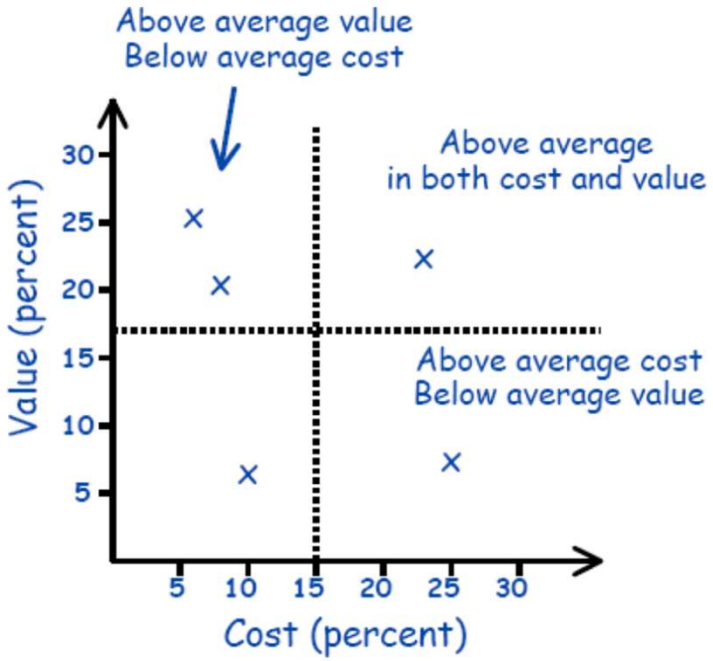
Req1 - 26% of the cost
Req2 - 50% of the cost
Req3 - 9% of the cost
Req4 - 16% of the cost

Plot ROI Graph

- Do AHP process **twice**:
  - Once to estimate relative value
  - Once to estimate relative cost

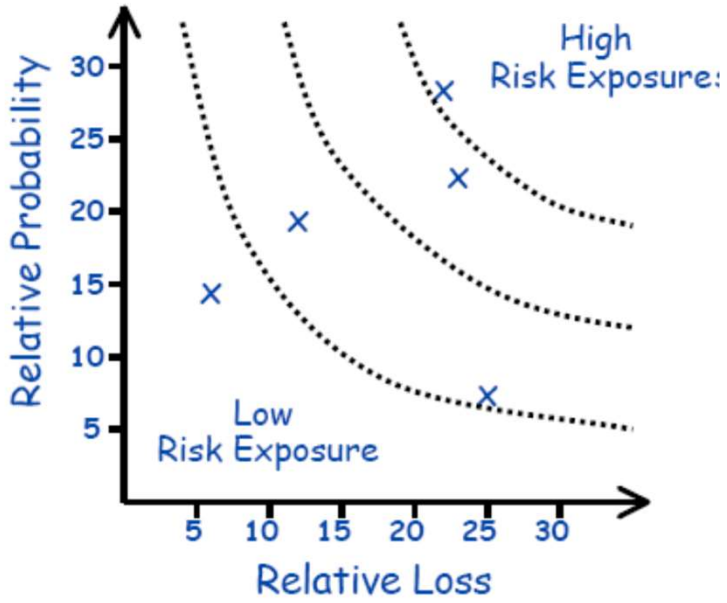
- Use results to calculate ROI ratio:



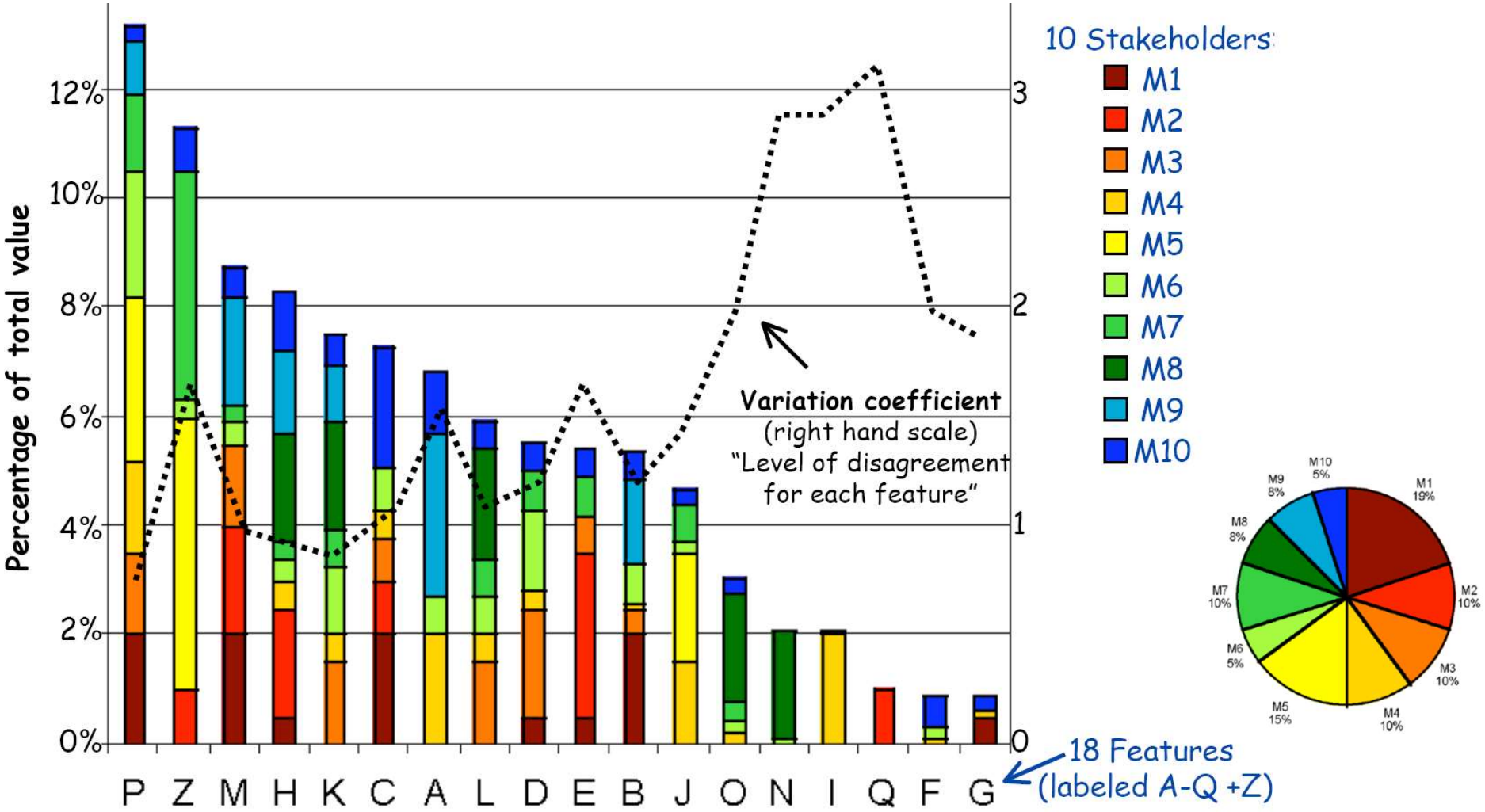


Other Selection Criteria

- ROI ratio is not the only way to group requirements
  - Risk = Loss vs. Probability
  - Etc.



Visualizing Value by Stakeholder



Basic Approaches to Conflict Resolution

- **Features of Stakeholder Conflict**
  - Deviant behavior & conflict are normal in small group decision making
  - More aggression and less co-operation when communication is restricted
  - Heterogeneous teams experience more conflict
  - Homogeneous groups are more likely to make high risk decisions (groupthink)
  - Effect of personality is overshadowed by situational and perceptual factors

- **Negotiation**
  - A collaborative exploration
    - Participants seek a settlement that satisfies all parties as much as possible.

- **Competition**
  - Maximizing your own gain:
    - No regard for the degree of satisfaction of other parties
    - But, not necessarily hostile

- **Third Party Resolution**
  - Participants appeal to outside source
    - judicial: cases presented by each participant are taken into account
    - arbitrary: e.g. toss of a coin

Considerations in Requirements Prioritization

- **Find factors that affects priority:**
  - How much does the customer want it?
  - How much cost to develop?
  - How much time to deliver?
  - How technologically difficult?
  - How organizationally difficult?
  - How much will the business benefit?

- **Not all factors apply to all projects**
  - Each factor's importance varies from project to project
  - '*Relative*' importance is different to everyone

- **Include all major stakeholders:**
  - We need to prioritize the requirements in collaboration with the customers and developers
  - We must decide on a subset of requirements to be first implemented among various stakeholder interests
  - We need to remember that more influence is exercised by a particular group of stakeholders

Requirements Prioritization Methods

- **Ranking**
  - When you rank requirements on an ordinal scale, you give each one a different numerical value based on its importance.
  - For example, the number 1 can mean that the requirement is the most important and the number n can be assigned to the least important requirement, n being the total number of requirements.

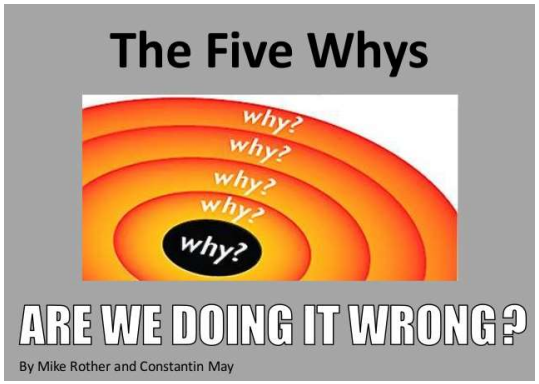
- **Numerical Assignment (Grouping)**
  - This method is based on grouping requirements into different priority groups with each group representing something stakeholders can relate to.
  - For example, requirements can be grouped into critical priority, moderate priority and optional priority.

- **MoScoW Technique**
  - Instead of numbers, this method uses four priority groups:
    - MUST (Mandatory)
    - SHOULD (High priority)
    - COULD (Preferred but not necessary)
    - WOULD (Can be postponed and suggested for future execution)

- **Bubble Sort Technique**
  - To prioritize requirements using bubble sort, you take two requirements and compare them with each other.
  - If you find out that one requirement should have greater priority over the other, you swap them accordingly. You then continue in this fashion until the very last requirement is properly sorted. The result is a list of requirements that are ranked.

- **Hundred Dollar Method**
  - This simple method is useful anywhere multiple stakeholders need to democratically vote on which requirements are the most important.
  - All stakeholders get a conceptual 100 dollars, which they can distribute among the requirements. As such, the stakeholder may choose to give all 100 dollars to a single requirement, or the person may distribute the points more evenly.
  - At the end, the total is counted and the requirements are sorted based on the number of points received.

- **Five Whys**
  - With five whys, the analyst asks the stakeholder repeatedly (five times or less) why the requirement is necessary until the importance of the requirements is established.
  - The answers reveal whether the requirement is really necessary or can be cancelled/postponed once the priority is determined.

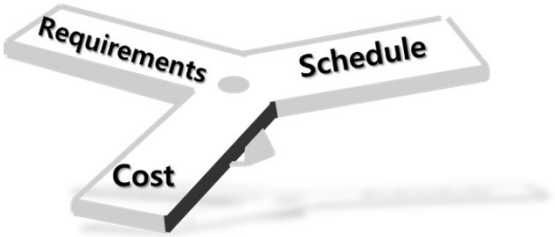


- **AHP** (Analytical Hierarchy Process)
  - Step 1. List all features and use cases that must be prioritized
  - Step 2. Estimate the relative benefit if each feature is included
  - Step 3. Estimate the relative penalty if each feature is not included
  - Step 4. Estimate the relative cost of implementing
  - Step 5. Estimate the relative degree of technical or other risk
  - Step 6. Calculate a priority number for each feature
  - Step 7. Sort the list of features

Relative Weights	2.0	1.0	1.0		0.5		0.5	
Feature	Relative Benefit	Relative Penalty	Relative Cost	Cost %	Relative Risk	Risk %	**Total Value**	**Value %**
Print a material safety data sheet	2	4	1	2.7	1	3.0	**8**	**5.2**
Query status of a vendor order	5	3	2	5.4	1	3.0	**13**	**8.4**
Generate a Chemical Stockroom inventory	9	7	5	13.5	3	9.1	**25**	**16.1**
See history of a specific chemical container	5	5	3	8.1	2	6.1	**15**	**9.7**
Search vendor catalogs for a specific chemical	9	8	3	8.1	8	24.2	**26**	**16.8**
Maintain a list of hazardous	3	9	3	8.1	4	12.1	**15**	**9.7**

Requirements Triage

- **Selecting the "right" features to include in next release**
  - Arriving at an answer is not easy.
  - It's either Win-Win or Lose-Lose.



- **Requirements vs. Schedule/Cost Risk**
  - **Basic triage**
    - An Engineering View
    - Balancing between requirements and Cost/Risk/Schedule
  - **Advanced triage**
    - A Business View
    - Balancing between requirements and Cost, Risk, Schedule, Market, Sales, Revenues, Pricing, Profit, and ROI

- **Tips for requirements triage**
  - **Maintain requirements** in **lists**
  - **Annotate requirements**, by at least relative priority and cost-to-satisfy
  - Don't ignore triage
  - Let schedule drive requirements inclusion
  - Involve representatives from **all key groups (stakeholders)**

Maintain Annotated Requirements Lists

- **Maintain sound advice to support all activities on requirements**
  - Enables you to answer questions such as:
    - How many requirements do you have?
    - How many high priority requirements do you have?
    - What percentage of the candidate requirements have you chosen to satisfy in your next release?
    - What percentage of the requirements deemed high priority by customer X are you satisfying?
    - If Sally quits, which requirements are affected?
    - What percentage of the requirements for this release have been validated?
    - And so on …
- **Find relevant importance to stakeholders**
  - What should we annotate?
    - Effect and cost
    - In which release?
    - Duration (optional)
    - Technical risk (optional)
- **Requirements should be in a database.**
  - Access, Excel, RequisitePro, **CaliberRM**, RTM, DOORS, etc.

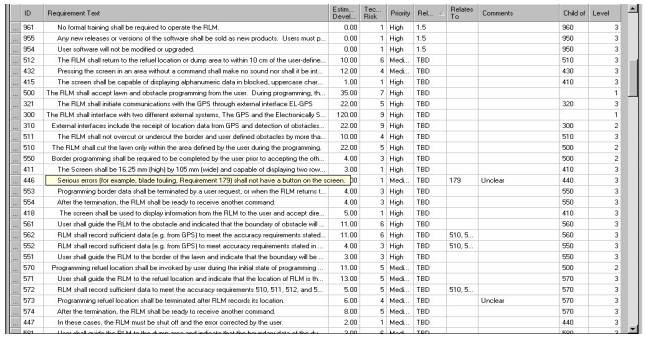
Annotate Requirements Example

We've ANNOTATED the features.

	ID	Requirement Text	Estim... Devel...	Tec... Risk	Priority	Rel...	Relates To	Comments	Child of	Level
...	961	No formal training shall be required to operate the RLM.	0.00	1	High	1.5			960	3
...	955	Any new releases or versions of the software shall be sold as new products. Users must p...	0.00	1	High	1.5			950	3
...	954	User software will not be modified or upgraded.	0.00	1	High	1.5			950	3
...	512	The RLM shall return to the refuel location or dump area to within 10 cm of the user-define...	10.00	6	Medi...	TBD			510	3
...	432	Pressing the screen in an area without a command shall make no sound nor shall it be int...	12.00	4	Medi...	TBD			430	3
...	415	The screen shall be capable of displaying alphanumeric data in blocked, uppercase char...	1.00	1	High	TBD			410	3
...	500	The RLM shall accept lawn and obstacle programming from the user. During programming, th...	35.00	7	High	TBD				1
...	321	The RLM shall initiate communications with the GPS through external interface EL-GPS	22.00	5	High	TBD			320	3
...	300	The RLM shall interface with two different external systems, The GPS and the Electronically S...	120.00	9	High	TBD				1
...	310	External interfaces include the receipt of location data from GPS and detection of obstacles...	22.00	9	High	TBD			300	2
...	511	The RLM shall not overcut or undercut the border and user defined obstacles by more tha...	10.00	4	High	TBD			510	3
...	510	The RLM shall cut the lawn only within the area defined by the user during the programming.	22.00	5	High	TBD			500	2
...	550	Border programming shall be required to be completed by the user prior to accepting the oth...	4.00	3	High	TBD			500	2
...	411	The Screen shall be 16.25 mm (high) by 105 mm (wide) and capable of displaying two row...	3.00	1	High	TBD			410	3
...	446	Serious errors (for example, blade fouling, Requirement 179) shall not have a button on the screen.	00	1	Medi...	TBD	179	Unclear	440	3
...	553	Programming border data shall be terminated by a user request, or when the RLM returns t...	4.00	3	High	TBD			550	3
...	554	After the termination, the RLM shall be ready to receive another command.	4.00	3	High	TBD			550	3
...	418	The screen shall be used to display information from the RLM to the user and accept dire...	5.00	1	High	TBD			410	3
...	561	User shall guide the RLM to the obstacle and indicated that the boundary of obstacle will ...	11.00	6	High	TBD			560	3
...	562	RLM shall record sufficient data (e.g. from GPS) to meet the accuracy requirements stated...	11.00	6	High	TBD	510, 5...		560	3
...	552	RLM shall record sufficient data (e.g. from GPS) to meet accuracy requirements stated in ...	4.00	3	High	TBD	510, 5...		550	3
...	551	User shall guide the RLM to the border of the lawn and indicate that the boundary will be ...	3.00	3	High	TBD			550	3
...	570	Programming refuel location shall be invoked by user during the initial state of programming ...	11.00	5	Medi...	TBD			500	2
...	571	User shall guide the RLM to the refuel location and indicate that the location of RLM is th...	13.00	5	Medi...	TBD			570	3
...	572	RLM shall record sufficient data to meet the accuracy requirements 510, 511, 512, and 5...	5.00	5	Medi...	TBD	510, 5...		570	3
...	573	Programming refuel location shall be terminated after RLM records its location.	6.00	4	Medi...	TBD		Unclear	570	3
...	574	After the termination, the RLM shall be ready to receive another command.	8.00	5	Medi...	TBD			570	3
...	447	In these cases, the RLM must be shut off and the error corrected by the user.	2.00	1	Medi...	TBD			440	3
...	581	User shall guide the RLM to the dump area and indicate that the boundary data of the du...	3.00	6	Medi...	TBD			580	3

The Result of Requirements Triage

- **An annotated list of requirements**
  - Requirements selected for inclusion are flagged.
  - Flagged requirements should be balanced with schedule and budget



ID	Requirement Text	Estim... Devel...	Tec... Risk	Priority	Rel...	Relates To	Comments	Child of	Level
961	No formal training shall be required to operate the RLM.	0.00	1	High	1.5			960	3
955	Any new releases or versions of the software shall be sold as new products. Users must p...	0.00	1	High	1.5			950	3
954	User software will not be modified or upgraded.	0.00	1	High	1.5			950	3
512	The RLM shall return to the refuel location or dump area to within 10 cm of the user-define...	10.00	6	Medi...	TBD			510	3
432	Pressing the screen in an area without a command shall make no sound nor shall it be int...	12.00	4	Medi...	TBD			430	3
415	The screen shall be capable of displaying alphanumeric data in blocked, uppercase char...	1.00	1	High	TBD			410	3
500	The RLM shall accept lawn and obstacle programming from the user. During programming, th...	35.00	7	High	TBD				1
321	The RLM shall initiate communications with the GPS through external interface EL-GPS.	22.00	5	High	TBD			320	3
300	The RLM shall interface with two different external systems, The GPS and the Electronically S...	120.00	9	High	TBD				1
310	External interfaces include the receipt of location data from GPS and detection of obstacles...	22.00	9	High	TBD			300	2
511	The RLM shall not overcut or undercut the border and user defined obstacles by more tha...	10.00	4	High	TBD			510	3
510	The RLM shall cut the lawn only within the area defined by the user during the programming.	22.00	5	High	TBD			500	2
550	Border programming shall be required to be completed by the user prior to accepting the oth...	4.00	3	High	TBD			500	2
411	The Screen shall be 15.25 mm (high) by 105 mm (wide) and capable of displaying two row...	3.00	1	High	TBD			410	3
446	Serious errors (for example, blade fouling, Requirement 179) shall not have a button on the screen.	00	1	Medi...	TBD	179	Unclear	440	3
553	Programming border data shall be terminated by a user request, or when the RLM returns t...	4.00	3	High	TBD			550	3
554	After the termination, the RLM shall be ready to receive another command.	4.00	3	High	TBD			550	3
410	The screen shall be used to display information from the RLM to the user and accept dire...	5.00	1	High	TBD			410	3
561	User shall guide the RLM to the obstacle and indicated that the boundary of obstacle will ...	11.00	6	High	TBD			560	3
562	RLM shall record sufficient data (e.g. from GPS) to meet the accuracy requirements stated...	11.00	6	High	TBD	510, 5...		560	3
552	RLM shall record sufficient data (e.g. from GPS) to meet accuracy requirements stated in ...	4.00	3	High	TBD	510, 5...		550	3
551	User shall guide the RLM to the border of the lawn and indicate that the boundary will be ...	3.00	3	High	TBD			550	3
570	Programming refuel location shall be invoked by user during the initial state of programming ...	11.00	5	Medi...	TBD			500	2
571	User shall guide the RLM to the refuel location and indicate that the location of RLM is th...	13.00	5	Medi...	TBD			570	3
572	RLM shall record sufficient data to meet the accuracy requirements 510, 511, 512, and 5...	5.00	5	Medi...	TBD	510, 5...		570	3
573	Programming refuel location shall be terminated after RLM records its location.	5.00	4	Medi...	TBD		Unclear	570	3
574	After the termination, the RLM shall be ready to receive another command.	8.00	5	Medi...	TBD			570	3
447	In these cases, the RLM must be shut off and the error corrected by the user.	2.00	1	Medi...	TBD			440	3

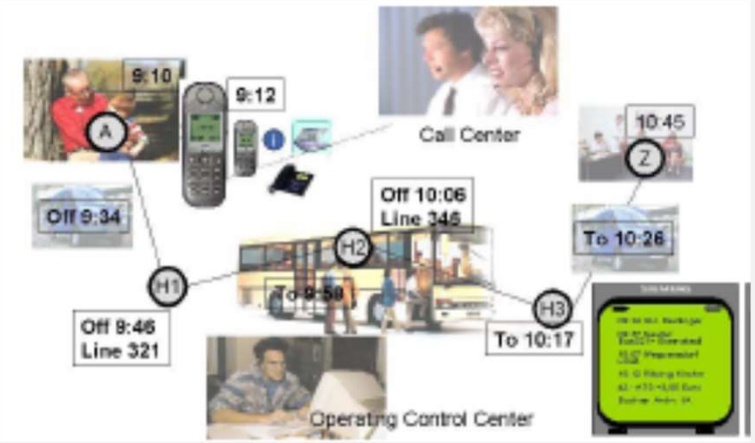
- All parties are in agreement

> - **All:** We agree that the following list of requirements is the best set of requirements for which we can now agree to, and represents the best balance between requirements, schedule, and budget.
> - **Marketing (or Customer Rep):** I agree to not change the requirements prior to product delivery.
> - **Development:** I agree to deliver this set of requirements with sufficient quality on this date: _____.
> - **Finance:** I agree to not reduce the total funding of this project below _____ .
> - **All:** We agree to work together to arrive at a new optimal solution in the event that any of us are forced to violate the above contracts.

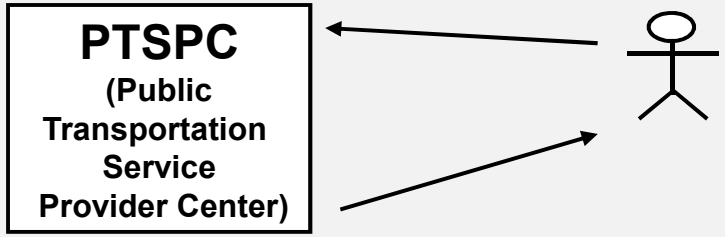
Exercise 3: Basic Prioritization & Selection

- **Prioritize the collected stakeholder requirements and select a subset of requirements doable within 3 months**
  - Project Title: "**Custom Mass Transportation System**" in 1990s
  - Purpose: Increase the usage ratio of regional/suburban mass transportation system

- **Justify your selection quantitatively**



Order mass transportation through SMS, call center, internet (Notify departing location, destination, time of departure/arrival, etc.)



Find optimal travel route, fare, ETA, and other traffic information

- 5 persons are available as workforce.
  - The project should be **under 15 man-month** if to complete within 3 months.
  - Assume that the Internet was booming.
- You have to add 2 columns of annotations → 4 of all.
  - Define each column precisely, including value and risk.
  - Use 4 different prioritization methods
  - Visualize your selection's rationale efficiently
  - Keep in mind the purpose of your selection

Requirements	Value (1~10)	Risk (1~10)			Effort (MM)	Selected (O / X)
Req. 1. The system should have features such as Register, Sign-in, Sign-out.					1	
Req. 2. The Driver should be able to view Passenger requests.					3	
Req. 3. The system should accept orders through the internet.					2	
Req. 4. The Customer should be able to designate the route in advance.					3	
Req. 5. The system should accept orders through SMS.					2	
Req. 6. The system should accept orders through the call center.					2	
Req. 7. Managers should be able to manage orders through the internet.					1	
Req. 8. Manager should be able to configure User profile through the call center.					2	
Req. 9. Data transfer between a taxi and traffic manager should be possible.					3	
Req. 10. Manager should be able to configure User profile through the internet.					1	