

Sequence Enumeration

This presentation summarises the excellent article By Rob Osana, to be found at

□ <http://www.embedded.com/2000/0009/0009feat2.htm>

There is a link from the course web page. Any errors in transcription are entirely mine :)

"Sequence enumeration is a technique for analyzing systems. It can be used to turn a complex set of requirements into an easily implemented state machine."

R. Osana

Sequence Enumeration: Requirements

Why do we need requirements?

- * To specify external system behavior
- * To specify implementation constraints
- * To serve as reference tool for maintenance
- * To predict future changes
- * To characterize responses to unexpected events

Sequence Enumeration: Requirements

The trouble is:

- System Developer must understand and organize them.
 - need technical background
 - must understand the user
 - must understand significance of each requirement
 - must understand priority of each requirement

- Written in natural language
 - Inaccurate
 - Ambiguous

What does "bridge" mean?

Sequence Enumeration: Bridges



Sequence Enumeration: Requirements 2

Requirements should be :

- Correct.
- Unambiguous.
- Complete.
- Consistent.

- Ranked for importance.
- Verifiable.
- Traceable.
- Modifiable.

Sequence Enumeration: Requirements 3

In spite of

- Client doesn't know what s/he wants
- Organizational factors
- Company politics
- new stakeholders

We need to

- reduce design complexity as it evolves
- determine all possible types of interaction with the system

Hence

- 'Use cases' to explore the interactions
- Sequence enumeration to specify the solution

Sequence Enumeration: Is

Sequence enumeration describes the external, or black box, behaviour of a system.

It is :

Theoretically and academically sound.

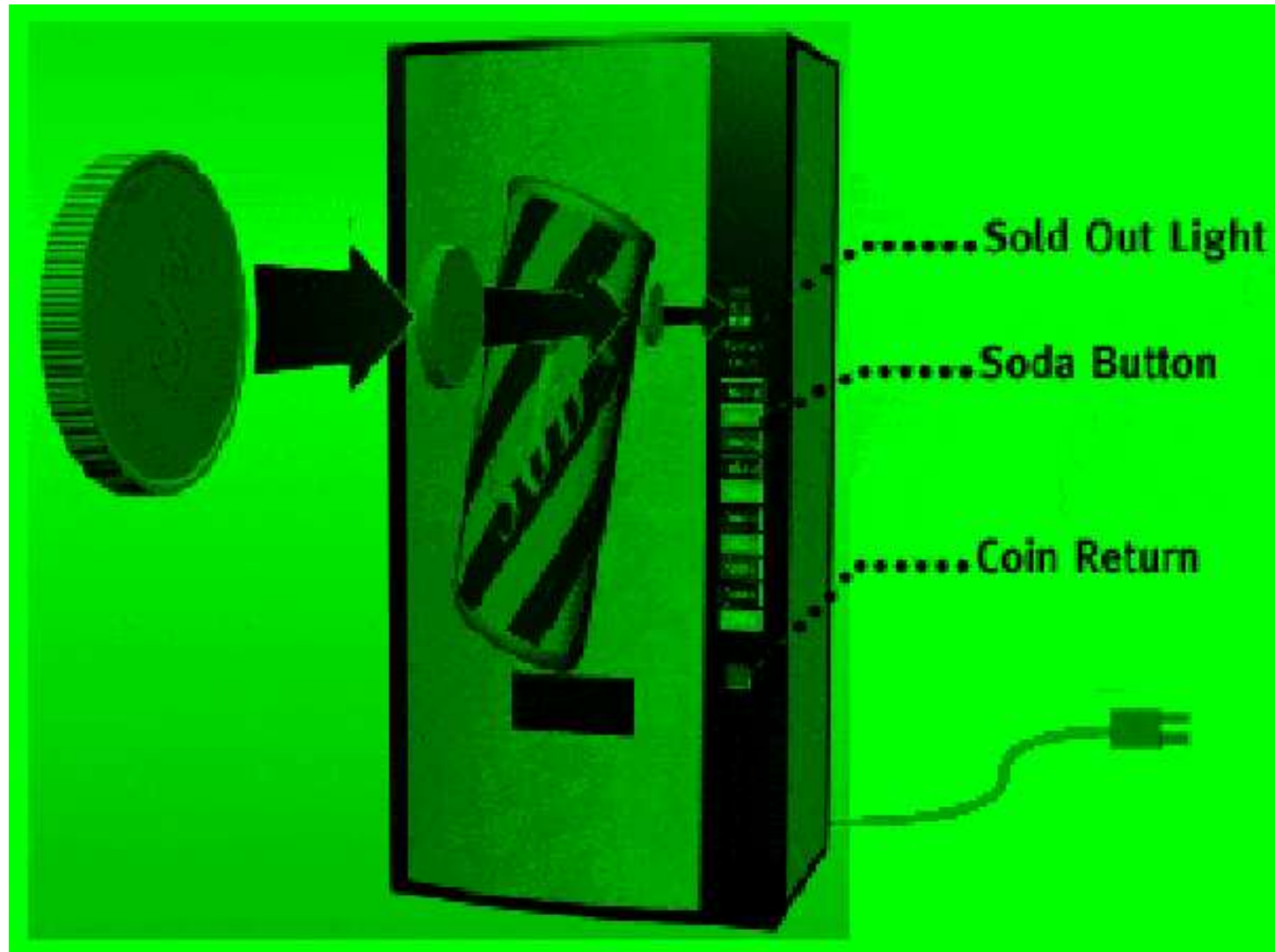
Highly practical.

How?

- Specify and map the stimuli and responses of the system.
- Build a list of prior stimuli + current stimuli
- Assess the response given the history of stimuli

Huh?

Sequence Enumeration: An example



Sequence Enumeration: Example 2

Natural Language Requirements :

- 1) Soda Machine will only produce a soda after 75 cents has been entered. The machine will only take quarters.
- 2) When the "Change Return" button is pressed, the available change in the input tray is returned.
- 3) When there is no more soda in the machine, the "Sold Out" light will be illuminated.

Sequence Enumeration: Use Case 1

Use Case:

A story about a system. It consists of :

- Preconditions
- Description
- Exceptions
- Illustrations
- Postconditions

Are good because they :

- Analyse and improve functional requirements.
- Model system functionality early on.
- Allow customer to understand the system operation.

Sequence Enumeration: Use Case 2

Precondition:

- Machine is ON and there is one soda in the machine.

Description:

- User walks up to machine and puts 2 quarters in.
- He then presses the soda button.
- Nothing comes out as 3 quarters are needed to get a soda.
- The user presses the change button and retrieves his change.
- User leaves to obtain another quarter.
- User returns, enters 3 quarters and selects a soda.
- Soda is delivered.
- The "sold out" light comes on as this was the last soda in the machine.

Exceptions:

- None.

Postconditions:

- There is no soda left in the machine and the sold out light has come on.

Sequence Enumeration: Enumeration

Table 1: Soda machine stimuli and response abbreviations

Stimulus Abbreviation

Initialize I

Quarter Q

Soda select SS

Change return CR

Sold out SO

Response Abbreviation

Soda deposited SD

Change return CHR

Sold out light SOL

Power on light LIGHT

Sequence Enumeration: Enumeration

Table 2: Enumeration level 1

Line Number	Enumeration Level	Stimuli Sequence	Response	Equivalence	Requirement
1	1	I	LIGHT		
2		Q	null	illegal	4
3		SS	null	illegal	none
4		CR	null	illegal	none
5		SO	null	illegal	none

Sequence Enumeration: Final State Machine

