

# New Visual Interface for Engineering Use Case Models

Nikolai Mansurov

Department for CASE tools,  
Institute for System Programming,  
Russian Academy of Sciences  
109004 Russia Moscow B.Kommunisticheskaya 25  
nick@ispras.ru

Dmitri Vasura

Department for CASE tools,  
Institute for System Programming,  
Russian Academy of Sciences  
109004 Russia Moscow B.Kommunisticheskaya 25  
vasura@ispras.ru

## Abstract

*This paper describes a new approach to visualization of scenarios within the use case-based engineering of functional requirements – the so-called Video Camera metaphor. The Video Camera metaphor facilitates involvement of business people, customers, problem domain experts and other non-technical stakeholders into capturing and validating formal requirements models. The key tool, supporting the Video Camera metaphor is the so-called Interface Editor which allows to draft the prototype user interface and automatically generates a user-friendly front-end to the set of formal modelling tools. The essence of the Video Camera interface is to associate sequences of events on a UML Sequence Diagram or an ITU-T Message Sequence Chart with sequences of activations of the elements of the generated user interface. The Video Camera interface allows capturing scenarios through direct activation of both the input and the output elements of the generated user interface. The generated user interface is also used to replay scenarios for validation purposes.*

## 1. Introduction

Department for CASE tools of the Institute for System Programming has been developing requirements engineering methodology [3,4], which is part of Accelerated Development Methodology (ADM) for specification, design, testing and re-engineering of telecommunication software [5]. Formal use case scenario models were selected as the notation for validation, transformation and analysis of functional requirements [3]. In this paper we describe a new visual interface to engineering functional requirements as use case scenarios. We believe that the new methods and tools will significantly improve adoption of formal modelling in industry.

The rest of the paper has the following organization. In section 2 we provide an overview of our requirements engineering methodology. In Section 3 we outline the key ideas of Video Camera visual interface. In section 4 we

outline the Use Case Studio toolkit, which implements the suggested approach.

## 2. Overview of our requirements engineering process

ADM requirements engineering process consists of several steps, which can be performed iteratively.

- Capture initial system requirements in tabular form, capture the set of external actors and use cases as UML use case diagrams;
- Design prototype user interfaces for all external actors using the Interface Editor tool, which automatically generates a user-friendly interface of the formal scenario model;
- Specify partial functional requirements by interactively capturing exemplary behavior for each use case by activating the elements of the generated user interface;
- Specify complete behavior of the system by identifying *episodes* (short interaction sequences, potentially down to individual *operations*) and arranging them into a UML Activity Diagram;
- Validate functional requirements using a combination of visual and formal techniques:
  - Replay validation scenarios against the generated user interface
  - Synthesize an executable model using the Event Automata approach [4], run model checking tool on the synthesized model [4], replay problematic sequences against the generated user interface.

After the complete requirements specification was specified and validated a draft design specification of the system in SDL and/or Promela and test cases in standard TTCN language are automatically synthesized from the use case model [4].

## 3. Video Camera interface to engineering scenario-based formal requirements models

In this section we outline the Video Camera approach to visualization of use case scenario models.

The Video Camera interface to formal scenario model is based on some powerful similarities between use-cases and *film industry*.

In film industry all events *occur* within some environment, while in the domain of engineering formal scenario-based models an algebraic notation is used to represent sequences of events. The essence of the Video Camera approach is to associate sequences of events with sequences of activations of the elements of the prototype user interface. This approach allows to "shoot" scenarios by *visually activating* both the input and the output elements of the user interface. The formal notation for event sequences can then be generated on the fly. The association between events in the formal model and visual activations of the elements of the generated user interface should work in both directions. Scenarios can be

(with notation for sequencing of episodes, alternatives and repetitions).

The advantage of the Video Camera interface is that all formal notations can be hidden from the requirements engineer; however, they are created on the fly and can be used by advanced users through conventional visual graphical tools. We believe, that building the interface of a requirements engineering CASE tool upon the "film industry" terminology can make it significantly more familiar to non-technical audience.

#### 4. UseCase Studio toolkit

UseCase Studio is a model-based requirements

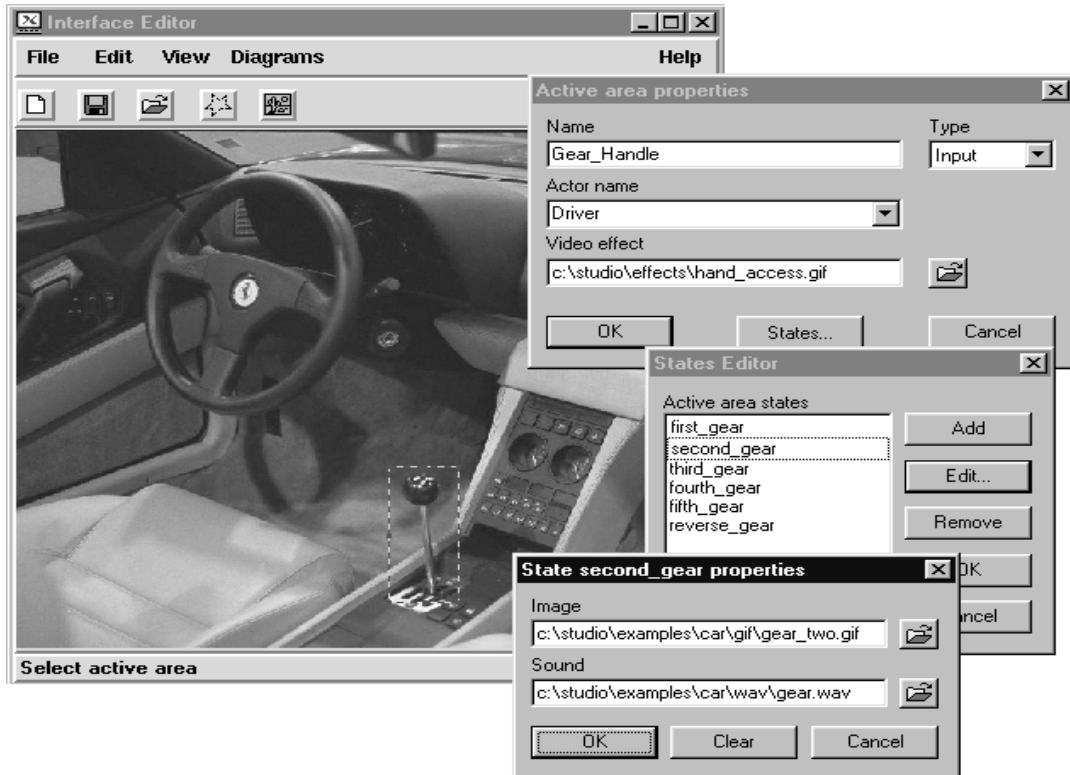


Figure 1. Interface Editor, updating selected active area

visualized by replaying them against the prototype user interface. With suitable use of animation, *illusion of the activation* of the input elements of the user interface can be created when visualizing scenarios.

In order for the Video Camera interface to function, the prototype user interface has to be explicitly created before scenarios can be captured. This is recommended by UML methodology, but was never enforced before.

Individual episodes can be identified, arranged into higher level scripts using UML Activity Diagram

engineering toolkit, which provides complete support for capturing and validating functional requirements with use case scenario models. The UseCase Studio implements the Video Camera metaphor for capturing and replaying use cases.

The UseCase Studio consists of the Validation and Code Generation Kernel, described in [4] and a set of visualization tools. The Kernel uses automatic synthesis technique to derive the so-called Event Automata from scenarios [3,4]. The Kernel is based on our MOST-SDL tool [3].

Visualization tools include the following:

- Use Case Editor - a visual editor for UML use cases
- Interface Editor - an interactive tool to create and edit Video Camera user interfaces;
- Scenario Recorder - Video Camera interface to the generated user interface with "Play", "Record", "Fast Forward" and "Fast Backward" buttons, similar to a Camcorder control panel;
- Episode Editor - a visual editor for UML Activity Diagrams or ITU-T High-Level Message Sequence Charts;
- Episode Simulator, - visual interface to simulation of UML Activity Diagrams;
- Sequence Diagram Editor - a visual editor for UML Sequence Diagrams or ITU-T Message Sequence

background has several *active areas*, the so-called user interface (UI) elements. Active areas can be visually arranged on the screen to create the desired layout. Input elements can be associated with an *activation effect*, which creates an illusion of a virtual user activating this element during playback (see Figure 2).

Generated user interface is used to capture scenarios, as well as to present traces for validation. Generated user interface is controlled by Scenario Recorder tool.

Association of UI elements with Sequence Diagram events has dual use: in "record" mode this association is used to create (edit) a Sequence Diagram; in "playback" mode this association is used to visualize a Sequence Diagram.

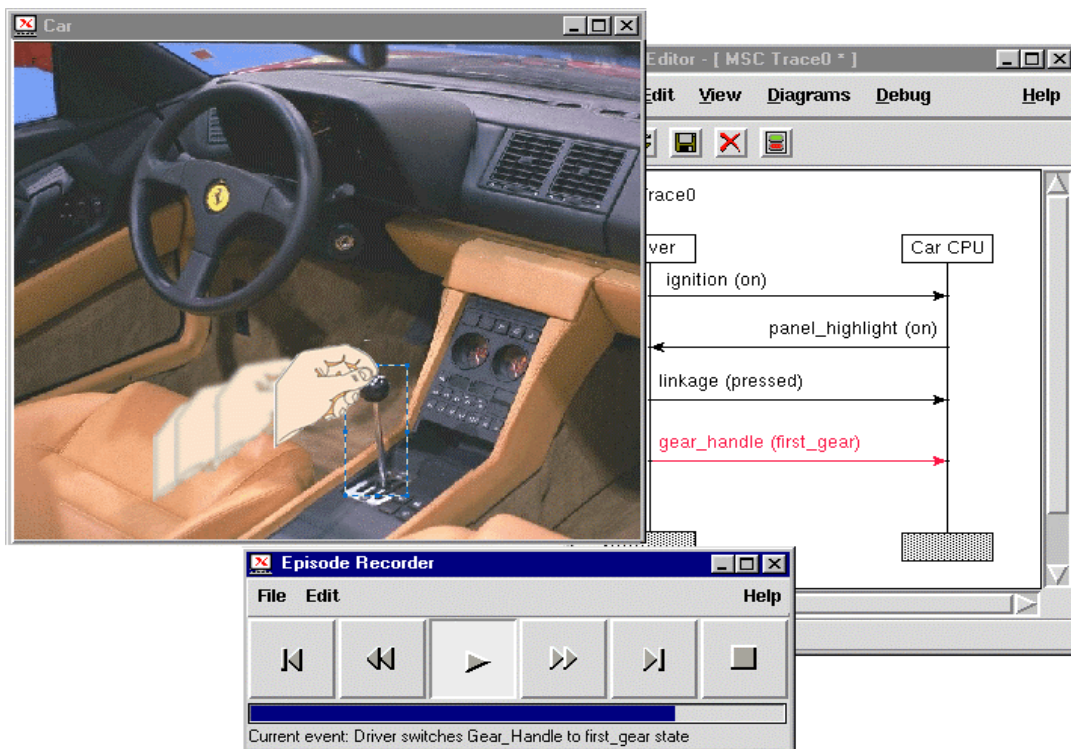


Figure 2. Scenario Recorder with the generated user interface in playback mode

- Charts;
- Model Navigator - visual access to the model-based repository.

#### 4.1. Interface Editor

Interface Editor of the UseCase Studio allows to visually and intuitively prototype a GUI, without having any experience in programming such interfaces for industry applications.

The editor looks like a usual graphical editor (see Figure 1). The prototype user interface consists of several *panels*. Each panel has a *background* (which can be imported from a digital camera, and edited). The

#### 4.2. Scenario Recorder

When all interfaces to actors for a certain use case are specified, scenarios for this use case can be recorded using Scenario Recorder Tool of the UseCase Studio.

Scenario Recorder tool allows to record sequences of events, corresponding to the behavior of external actors and the system by visually activating (i.e. pressing, selecting, switching etc.) UI elements of user interface panels. Sequences of events activated through UI elements can be simultaneously displayed in the Sequence Diagram Editor.

Scenario Recorder panel looks like a Camcorder control panel (see Figure 1). Using “back” and “fast back” buttons from the panel user may back for some steps and then start recording again and replay the former sequence of actions using buttons “forward” and “fast forward”. Both back and forward operations are reflected on the Sequence Diagram.

During simulation all events traversed by the Scenario Recorder are visualized against user interfaces panels as changing states of UI elements. Activation effects create an illusion of activation of the UI elements by a virtual user.

### 4.3. Episode Editor and Episode Simulator

When all scenarios for a certain use case are recorded in Scenario Recorder, the Episode Editor can be used to combine these scenarios into an Activity Diagram, which specifies complete behaviour of the system in the given use case.

When the Activity Diagram is completed, the use case can be simulated using the Episode Simulator tool of the UseCase Studio. User Interfaces for all actors, which use system in this use case, are started and Episode Simulator highlights the start node of the Activity Diagram.

The Episode Simulator also has a Camcorder control panel. Step-by-step simulation of the Activity Diagram can be performed using “forward”, “play” and “back” buttons on the Episode Simulator panel. When “play” button is pressed, simulator steps through the Activity Diagram, i.e. makes a transition from the currently selected symbol to the next symbol by the connectivity line. If there are several possible transitions from the currently selected symbol, Episode Simulator highlights all possible symbols, to which the transition may be done, and asks the user to make a choice.

Whenever a reference symbol is reached, the corresponding Sequence Diagram is loaded into the Sequence Diagram Editor and simulation is continued in the Scenario Recorder.

### 5. Related work

Visualization of formal specifications is becoming an active research field [2]. Message Sequence Charts [2] is probably the most user-friendly formal specification notation. Several research groups are using MSCs as “front-end” formal specification language to automatically synthesize validation specifications in more mathematical-oriented (and thus, having a higher accessibility barrier) languages, such as Promela [6].

However, visualization provided by MSC and SDL tools does not do beyond the visual syntax of the specification language itself. Therefore, MSC and SDL requirements specifications are captured in internal terms

of the formal specifications domain, rather than in external, not-technical terms.

A more advanced approach is reported in [7]. AMBER is a visual formal notation for describing and analysing business processes models [7]. The TestBed Studio provides a non-technical GUI for visualizing, creating and simulating the model.

### 6. Conclusions

In this paper we described a new visual interface to engineering functional requirements as use cases, the so-called Video Camera metaphor. Approach presented in this paper is an extension of the ADM requirements engineering methodology. The new approach is build around several powerful similarities, which exist between scenario modelling and film industry.

The Video Camera interface is implemented in our Use Case Studio toolkit. The Video Camera interface is used as a front-end to the validation and code generation kernel.

In our opinion, use case scenario model is a very attractive visual formal notation, which on one hand allows to perform validation and transformation, and on the other hand, allows user-friendly visualization. We believe, that the suggested approach together with a set of analysis, transformation and validation tools produces a comprehensive requirements engineering environment, which can be used by non-technical experts.

### 7. References

1. 1<sup>st</sup> Int. Symposium on Visual Formal Methods (VFM'99), Ed. S. Mauw, et.al., Eindhoven, The Netherlands, 1999, Computing Science Reports, Department of Mathematics and Computing Science, Eindhoven University of Technology, report 99-08
2. ITU-T, Rec. Z.120, Message Sequence Charts (MSC), Geneva, 1996
3. N. Mansurov, Automatic Synthesis of SDL from MSC in Forward and Reverse Engineering, in [2], pp. 44-64
4. N. Mansurov, D. Vasura, Approximation of (H)MSC semantics by Event Automata, SAM'2000 workshop, Grenoble, France, 2000
5. R. Probert, N. Mansurov, Improving time-to-market using SDL tools and techniques (tutorial), in Proc. 9<sup>th</sup> SDL Forum, Montreal, Canada, June 21-26, 1999
6. H. Ben-Abdallah, S. Leue, MESA: Support for Scenario-Based Design of Concurrent Systems, in Proc. TACAS'98, Lisbon, Portugal, 1998, pp. 118-135
7. P. Luttinghuis, Visualizing Business Processes, in [2], pp 82-113