

# Generating XML-DITA Procedures From Use Case Detection In Informal Technical Documents



**DITA Conference, Munich 2009, November 16th-17th**  
**Michel Lanque, Philippe Larvet**

# Context

---

- **Technical & Customer Documentation**
- **Objective** : to help technical documents **writers** (System & Development teams, documentation writing teams) to build “procedures” from the contents of **legacy** technical specifications.
- These “procedures”, which have a **technical** origin, are dedicated to be used to write **customer documentation**.

# Information Source and Content Management

---

## Sources

- Technical knowledge
- System specifications documents

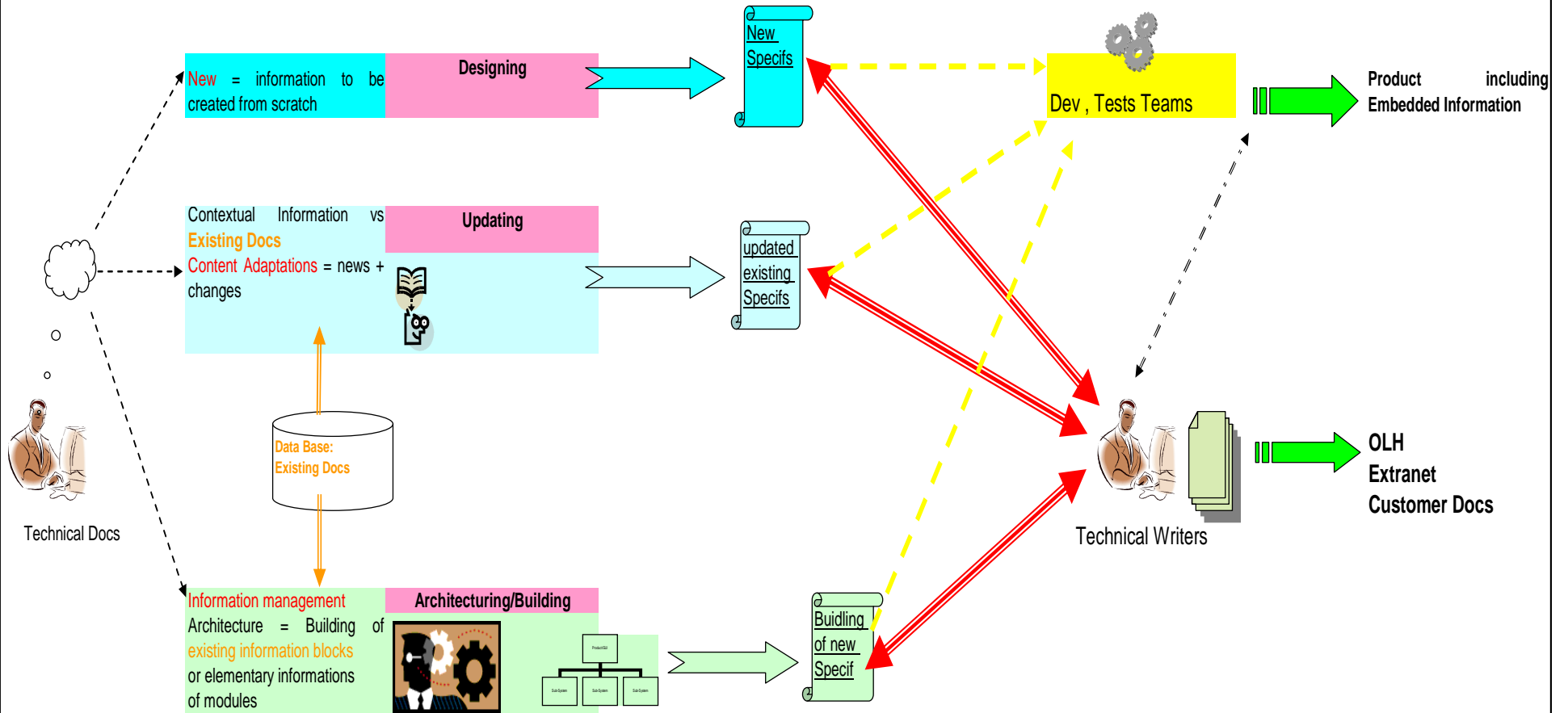
## Key inputs for users

- Use cases
- Informal Procedures

## Content management in Authoring

- Legacy information management
- Concept of Use Case Detection system
- R&D about the Detection system and content management

# Information source management



# Source Information Engineering to User Functional Information System

---

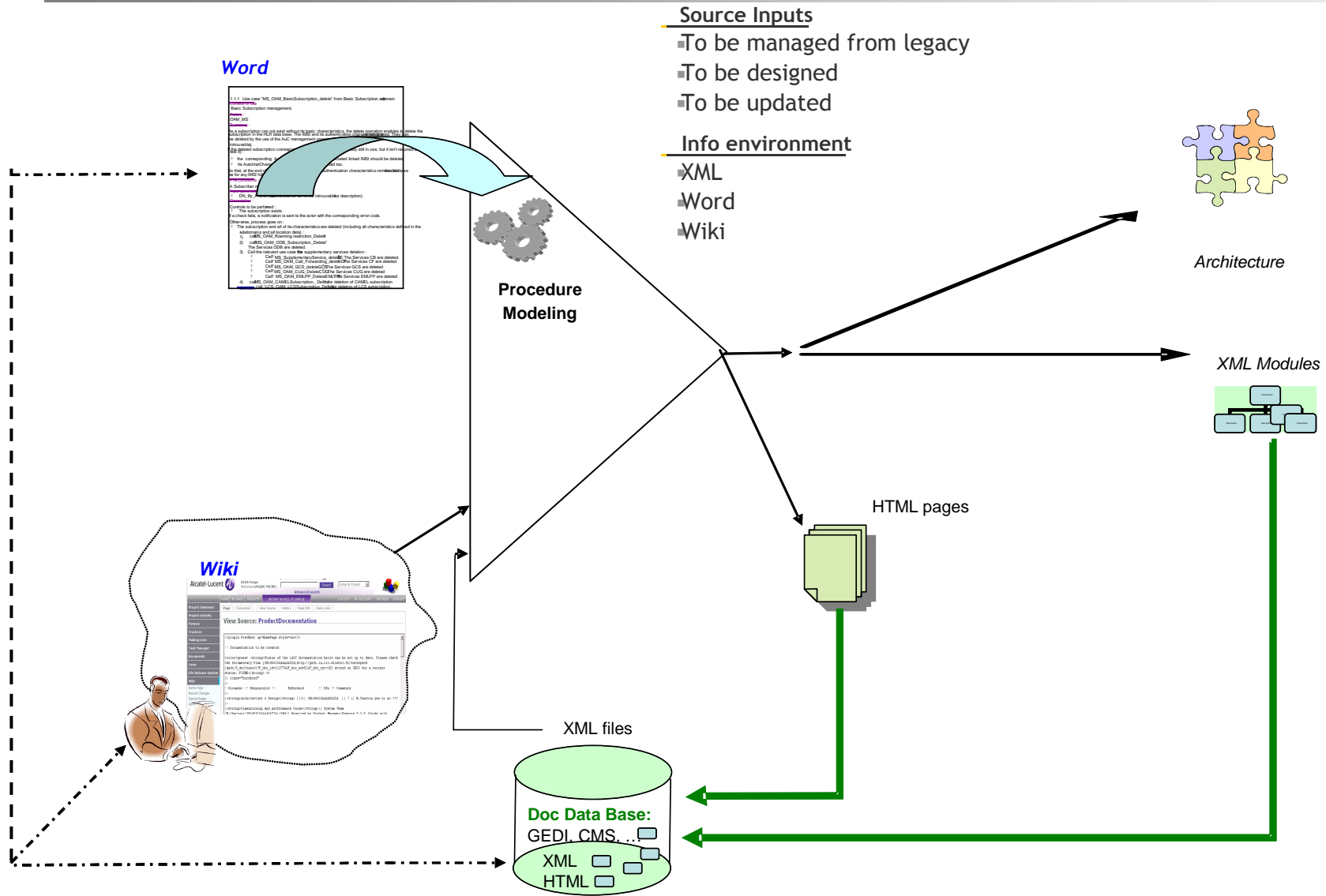
## Operator Information System

- Documentation, Extranet, embedded information through GUIs, OnLine Helps, ...  
=> provide all information to customers/end-users enabling them to operate & maintain their products.
- Key Customer Information => mainly related to functional procedures and maintenance.
- A part of systems Efficiency and Performance relies on customer information access, using it for (re)-action of system.

## Single source Authoring system

- Technical specifications processing (use cases)
  - Source for operator information systems (user procedures)
  - Content management DITA compliant

# Source Inputs And Target Process



# The main question

---

Technical specifications are not usable as it is for customer information :

- not customer-oriented,
- not complete
- not always available, etc.



How to transform pieces of technical specifications  
into « operator procedures » for Customer  
Documentation ?

# Constraints

---

- Use of **legacy** informal documents (rewrite existing documentation as formal procedures is not realistic)
- Concrete results are expected within a **short delay**
- No **special training** on Products has to be given to delocalized Technical writers
- **Standard XML** documentation modules (**DITA-compliant**) are expected to be stored in a modern XML-based Documentation System



## Our way of research

---

- Observation: Legacy Technical Documents are mainly written by using **Use Cases**
- Procedures can be **formally** expressed from Use Cases
- **Extracting Use Cases** from technical documents and **expressing formally** these use cases would help to prepare writing of operator procedures

## How to do it ?

---

- Extract **use cases structure** and different parts of **use cases contents**
- Build **formal procedures** by **re-arranging** these contents
- Generate **XML-DITA modules** from formal procedures
- Automate this process through a **dedicated tool** to help writing teams

## Extracting Use Cases structure & contents

---

- **Text analysis** of the technical document
- **Pertinent words** extraction
- **Keyword recognition** via a dedicated **ontology** ('Actor', 'Pre-conditions', 'Operations', etc.)
- Extraction of corresponding **paragraphs** : each paragraph is a **use case element**
- Measuring **semantic distance** between paragraphs to be able to **re-assemble** them later within another structure

## Building Formal Procedures

---

- **Gathering** all the elements for a given use case, by taking into account the **semantic distance** between the paragraphs within the original text
- **Reorganizing** the elements according to a given **pattern** = building automatically the **structure of a procedure** from the distinct paragraphs.
- => For each use case (UC), extracted **paragraphs** are **organized** in order to build the **structure** of the corresponding **procedure**.
- *For example, paragraphs "Actors" and "Summary" of the UC can be concatenated as part "Context" of the procedure*
- *Other example, the paragraph "Pre-conditions" of the UC can be used to build the part "Input parameters" of the procedure, etc.*

## Generating XML-DITA Modules

---

- Finally, an **XML version** of the procedure can be generated, according to DITA-compliant XML schemas
- XML is **natively** used by our CMS (**Content Management Documentation System**)
- DITA is now considered an international **standard**

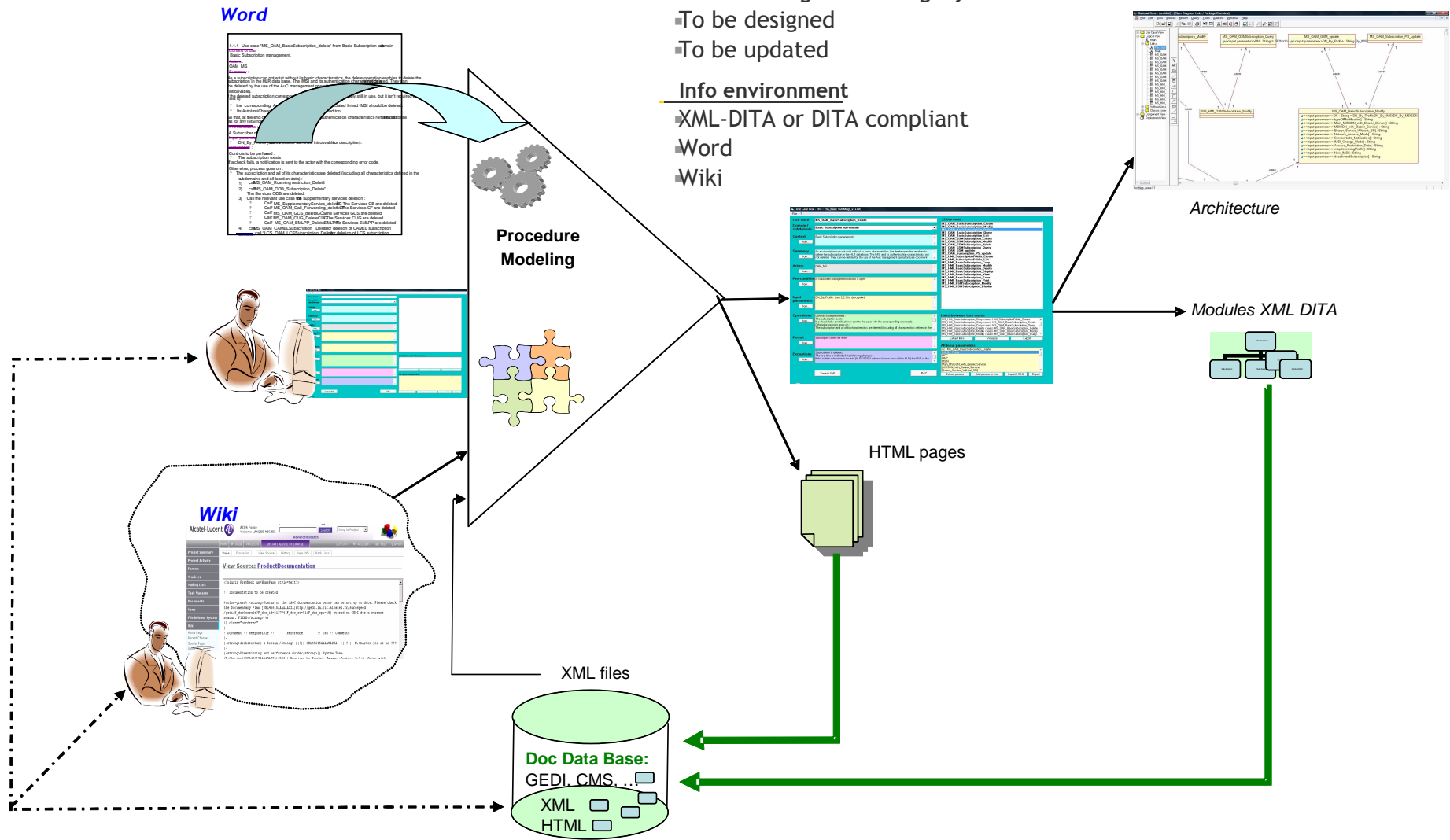
# Automating The Process With A Tool

## Source Inputs

- To be managed from legacy
- To be designed
- To be updated

## Info environment

- XML-DITA or DITA compliant
- Word
- Wiki



# The Tool : Procedure Modeler

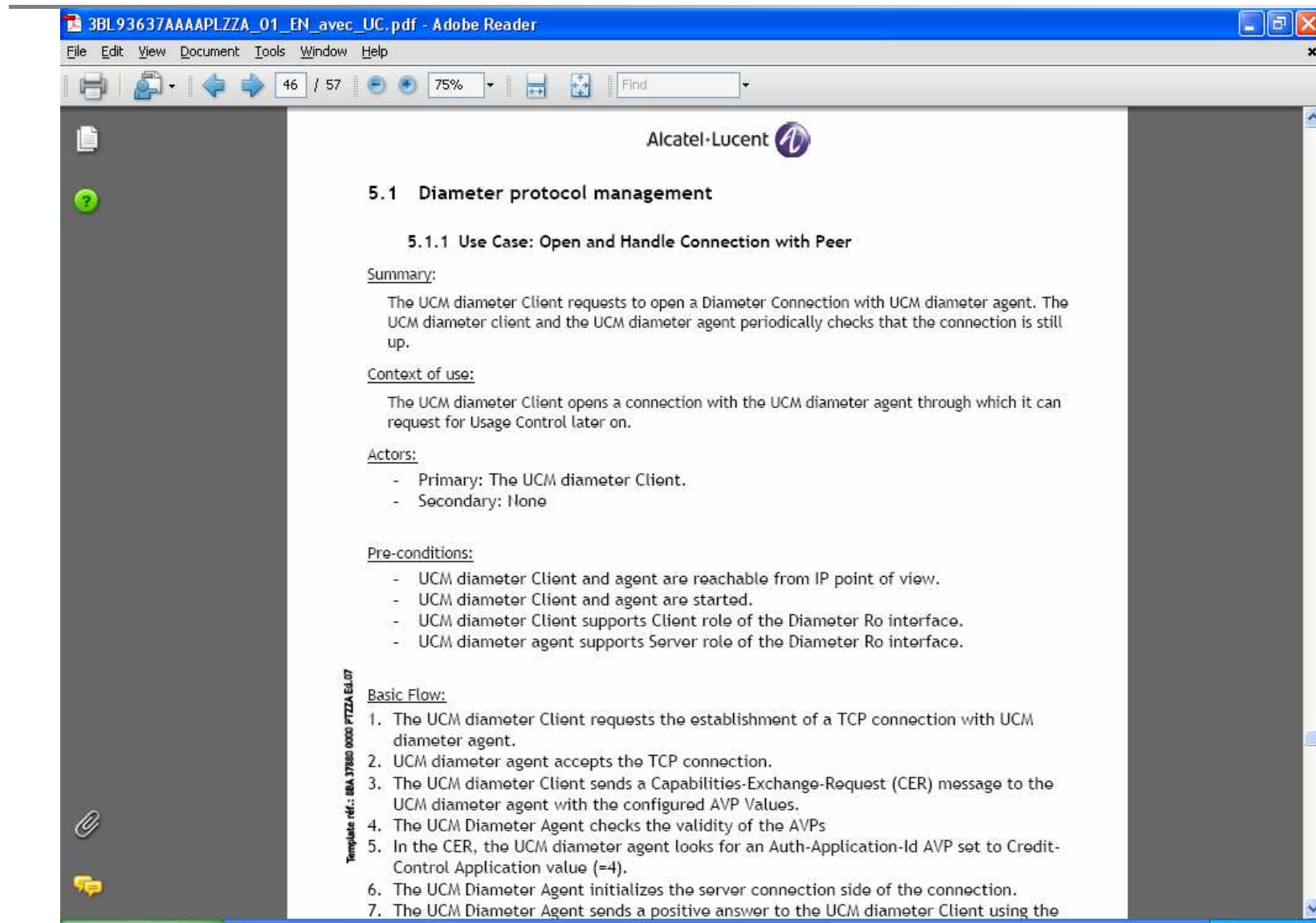
The screenshot displays the 'Procedure Modeler' software interface. The main window is titled 'Use Case Box - TRS : TRS\_Basic SubMngt\_v11.bt'. The interface is divided into several sections:

- Use case:** MS\_OAM\_BasicSubscription\_Delete
- Domain / subdomain:** Basic Subscription sub-domain
- Context:** Basic Subscription management.
- Summary:** As a subscription can not exist without its basic characteristics, the delete operation enables to delete the subscription in the HLR data base. The IMSI and its authentication characteristics are not deleted. They can be deleted by the use of the AuC management operation (see document)
- Actors:** OAM\_MS
- Pre-condition:** A Subscriber management session is open
- Input parameters:** DN\_By\_Profile : [see 2.2.3 for description]
- Operations:** Controls to be performed : The subscription exists. If a check fails, a notification is sent to the actor with the corresponding error code. Otherwise, process goes on : The subscription and all of its characteristics are deleted (including all characteristics defined in the
- Result:** subscription does not exist
- Exceptions:** Subscription is deleted. The real time is notified of the following changes : If the mobile subscriber is located [VLR/ SGSN address known and valid in HLR] the VLR or the

On the right side, there is a list of 22 Use cases, with 'MS\_OAM\_BasicSubscription\_Delete' highlighted. Below this list are sections for 'Links between Use cases' and 'All Input parameters'.

Buttons at the bottom include 'Save in XML', 'RAZ', 'Extract links', 'Visualize', 'Export', 'Extract params', 'Add params to visu', 'Impact HTML', and 'Export'.

# Example of a Use case within a Technical Requirement Specification



The screenshot shows a PDF document titled "3BL93637AAAAPLZZA\_01\_EN\_avec\_UC.pdf" in Adobe Reader. The document is from Alcatel-Lucent and details a use case for Diameter protocol management. The use case is titled "5.1.1 Use Case: Open and Handle Connection with Peer". It includes sections for Summary, Context of use, Actors, Pre-conditions, and Basic Flow.

**Alcatel-Lucent**

## 5.1 Diameter protocol management

### 5.1.1 Use Case: Open and Handle Connection with Peer

Summary:

The UCM diameter Client requests to open a Diameter Connection with UCM diameter agent. The UCM diameter client and the UCM diameter agent periodically checks that the connection is still up.

Context of use:

The UCM diameter Client opens a connection with the UCM diameter agent through which it can request for Usage Control later on.

Actors:

- Primary: The UCM diameter Client.
- Secondary: None

Pre-conditions:

- UCM diameter Client and agent are reachable from IP point of view.
- UCM diameter Client and agent are started.
- UCM diameter Client supports Client role of the Diameter Ro interface.
- UCM diameter agent supports Server role of the Diameter Ro interface.

Basic Flow:

1. The UCM diameter Client requests the establishment of a TCP connection with UCM diameter agent.
2. UCM diameter agent accepts the TCP connection.
3. The UCM diameter Client sends a Capabilities-Exchange-Request (CER) message to the UCM diameter agent with the configured AVP Values.
4. The UCM Diameter Agent checks the validity of the AVPs
5. In the CER, the UCM diameter agent looks for an Auth-Application-Id AVP set to Credit-Control Application value (=4).
6. The UCM Diameter Agent initializes the server connection side of the connection.
7. The UCM Diameter Agent sends a positive answer to the UCM diameter Client using the

Template ref.: 3BA 37850 0000 PZZA 01.07



# Processing this text with Procedure Modeler

The screenshot shows the Procedure Modeler interface. The main window displays a hierarchical tree structure of technical requirements:

- Technical Req. Spec.
  - 8611 USAGE CONTROL DIAMETER AGENT EXTERNAL DIAMETER INTERFACE
    - Chapter 3 DIAMETER PROTOCOL
    - Chapter 1 INTRODUCTION
    - UC\_Chapter 5 USE CASES
    - Chapter 6 DISTRIBUTION AND REDUNDANCY

A pop-up window titled "Open and Handle Connection with Peer" is open, showing the following details:

**Edit**

- \* TRS\_8611 Usage Control-Diameter Agent External Diameter Interface. Chapter 5. USE CASES
- \* Procedure Open and Handle Connection with Peer

**c:** Open and Handle Connection with Peer  
**p=** 0  
**d=** 04-09-2009

**o:** operation

**Summary:**  
 The UCM\_diameter\_Client requests-to-open a Diameter Connection.  
 The UCM\_diameter\_Client periodically-checks the connection.  
 The Diameter Connection is-opened-with UCM\_diameter\_agent.  
 The UCM\_diameter\_agent periodically-checks that the connection is still up.

**Context of use:**  
 \* The UCM diameter Client opens a connection with the UCM diameter agent through which it can request for Usage Control later on.

**Actors:**  
 The UCM\_diameter\_Client is a primary Actor.  
 \* No Secondary Actor.

**Pre-conditions:**  
 \* - UCM diameter Client and agent are reachable from IP point of view.  
 \* - UCM diameter Client and agent are started.  
 UCM\_diameter\_Client supports-Client-role-of the Diameter Ro interface.  
 UCM\_diameter\_agent supports-Server-role-of the Diameter Ro interface.

**Operations:**  
 Description of data

Buttons: Save, Analyze, Proc Modeler, Cancel

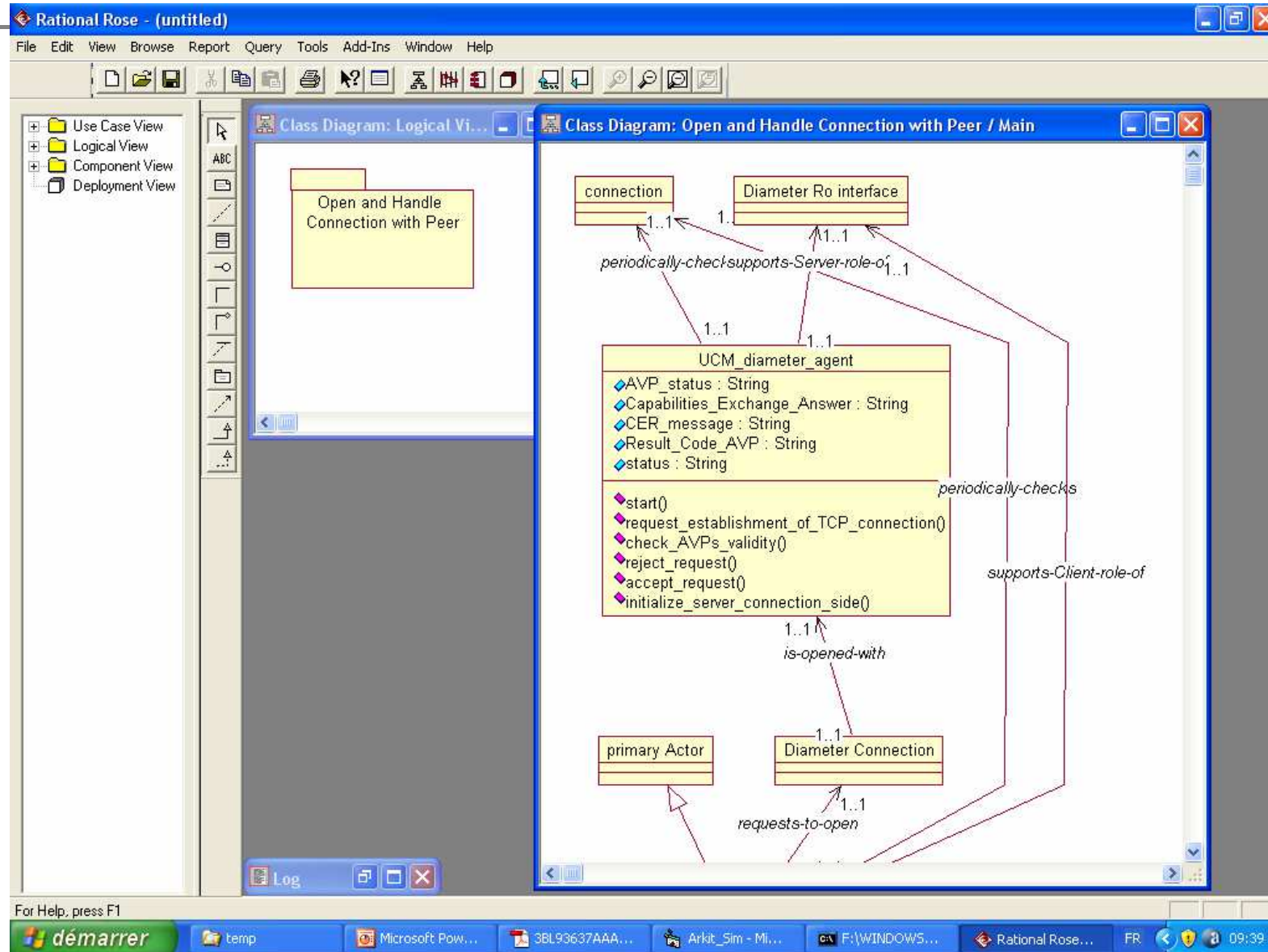
# Natural-Language analysis, Object model building

The screenshot displays a software development environment with the following components:

- Technical Req. Spec.:** 8611 USAGE CONTROL DIAMETER AGENT EXTERNAL DIAMETER INTERFACE
- Chapter 3:** DIAMETER PROTOCOL
- Chapter 1:** INTRODUCTION
- UC\_Chapter 5:** USE CASES
- Open and Handle Connection with Peer:** A window showing a list of use cases:
  - Open and Handle Connection with Peer
  - Session fail over
  - Peer Disconnection
  - Usage Control for event base
  - Usage Control for session base
- Results of NL Analysis:** A window showing the results of natural language analysis for the selected use case:
  - Classes:** connection, Open and Handle Connection with Peer, Diameter Connection, Open and Handle Connection with Peer, Diameter Ro interface, Open and Handle Connection with Peer, primary Actor, Open and Handle Connection with Peer, UCM\_diameter\_agent, Open and Handle Connection with Peer, UCM\_diameter\_Client, Open and Handle Connection with Peer
  - Operations:** UCM\_diameter\_agent supports-Server-role-of the Diameter Ro interface.
  - Description of data:** (Empty)

The Windows taskbar at the bottom shows the following applications: démarrer, Arkit\_Sim, Microsoft PowerPoint..., 3BL93637AAAAPLZZ..., 4 Visual Basic, FR, and the system clock at 09:26.

# Object model graphical representation



# Object model validation through simulation

The screenshot shows a simulation window titled "Simulator. Architecture : Open and Handle Connection with Peer. Subsystem : Open and Handle Connection with Peer". The window is divided into several panes:

- GUI:** Contains sections for "Status" (displaying "\*\*\* Operation in progress : UCM\_diameter\_agent.request\_establishment\_of\_TCP\_connection"), "Message" (displaying "UCM diameter agent accepts the TCP connection"), "Dialog with the User", and "Input Data".
- Internal View:** A light blue area displaying object attributes for "UCM\_diameter\_agent" (AVP\_status, Capabilities\_Exchange\_Answer, CER\_message, Result\_Code\_AVP, status) and "UCM\_diameter\_Client" (answer, message\_sent, status).
- Caution! Dialog Box:** A central dialog box with a red 'X' icon. It lists relationships for 'UCM\_diameter\_agent':
  - (1..1) periodically-checks (1..1) connection
  - (1..1) supports-Server-role-of (1..1) Diameter Ro interfaceIt states: "'UCM\_diameter\_agent' has no relationship with 'UCM\_diameter\_Client'" and asks: "Would you add this relationship to 'UCM\_diameter\_agent' ?". It has "Oui" and "Non" buttons.
- Execution stack:** A light blue area at the bottom left.
- Navigation:** "Next step", "Fast |", and "Simulation Log" buttons.

The Windows taskbar at the bottom shows the "démarrer" button, a "temp" folder, and several open applications including "Microsoft Pow...", "3BL93637AAA...", "5 Visual Basic", "F:\WINDOWS...", "Rational Rose...", and "FR". The system clock shows "09:44".

# Object model validation through simulation

The screenshot shows a simulation window titled "Simulator. Architecture : Open and Handle Connection with Peer. Subsystem : Open and Handle Connection with Peer". The window is divided into several panes:

- GUI:** Shows a status message: "\*\*\* Operation in progress : UCM\_diameter\_agent.request\_establishment\_of\_TCP\_connection".
- Internal View:** Displays a list of objects and their attributes, such as "UCM\_diameter\_agent" with attributes like "AVP\_status=", "Capabilities\_Exchange\_Answer=", "CER\_message=", "Code\_AVP=", "Diameter\_Client", "Message\_sent=", and "Server\_role\_of\_Diameter\_Ro\_interface=".
- Dialog with the User:** A dialog box with a red 'X' icon. It lists existing relationships for 'UCM\_diameter\_agent':
  - (1..1) periodically-checks (1..1) connection
  - (1..1) supports-Server-role-of (1..1) Diameter Ro interfaceIt then states: "'UCM\_diameter\_agent' has no relationship with 'UCM\_diameter\_Client'". Below this, it asks: "Would you add this relationship to 'UCM\_diameter\_agent' ?". There are "Oui" and "Non" buttons.
- Execution stack:** A pane at the bottom left, currently empty.
- Navigation buttons:** "Next step", "Fast I", and "Simulation Log" buttons are located at the bottom right.

A blue callout box with white text is overlaid on the dialog box, stating: "Here, for instance, a relationship is missing between two objects. The simulator detects it and proposes to automatically enrich the model." A blue arrow points from the callout box to the dialog box.

# Generating the procedure after validation

The screenshot displays a software interface for a simulation. The main window is titled "Simulator, Architecture : Open and Handle Connection with Peer. Subsystem : Open and Handle Connection with Peer". It features a menu bar with "Simulation", "Scenarios", "Trace", "Covering", and "Current action". The interface is divided into several panels: "GUI" on the left, "Internal View" on the right, and "Execution stack" at the bottom. A central dialog box titled "Open and Handle Connection with Peer\_gen" is open, showing a procedure definition. The dialog box contains the following text:

```
-----  
* Procedure Open and Handle Connection with Peer  
-----  
c: Open and Handle Connection with Peer  
p= 0  
d= 04-09-2009  
  
o: operation  
-----  
Context:  
* The UCM diameter Client opens a connection with the UCM diameter agent through  
which it can request for Usage Control later on.  
  
Summary:  
Diameter Connection is-opened-with one UCM_diameter_agent  
UCM_diameter_agent periodically-checks one connection  
UCM_diameter_agent supports-Server-role-of one Diameter Ro interface  
UCM_diameter_agent uses some UCM_diameter_Client  
UCM_diameter_Client periodically-checks one connection  
UCM_diameter_Client requests-to-open one Diameter Connection  
UCM_diameter_Client supports-Client-role-of one Diameter Ro interface  
UCM_diameter_Client uses some UCM_diameter_agent  
  
Actors:  
UCM_diameter_Client is a kind of primary Actor  
  
Input parameters:  
UCM_diameter_agent has the following data:  
- AVP_status:String  
- Capabilities_Exchange_Answer:String  
- CER_message:String  
- Result_Code_AVP:String  
- status:String
```

The dialog box has "Save", "Proc Modeler", and "Cancel" buttons at the bottom. The "Execution stack" panel shows the following sequence of actions:

```
UCM_diameter_Client->operation : connect  
UCM_diameter_Client->dialog "A Capabili  
UCM_diameter_Client->message_sent="configured AVP Values"  
UCM_diameter_Client->UCM_diameter_agent.accept_CER_message(UCM_diameter_Client.message_sent)
```

The Windows taskbar at the bottom shows the "démarrer" button, a "temp" folder, and several open applications including "Microsoft Pow...", "3BL93637AAA...", "Visual Basic", and "Rational Rose...". The system clock shows "09:47".

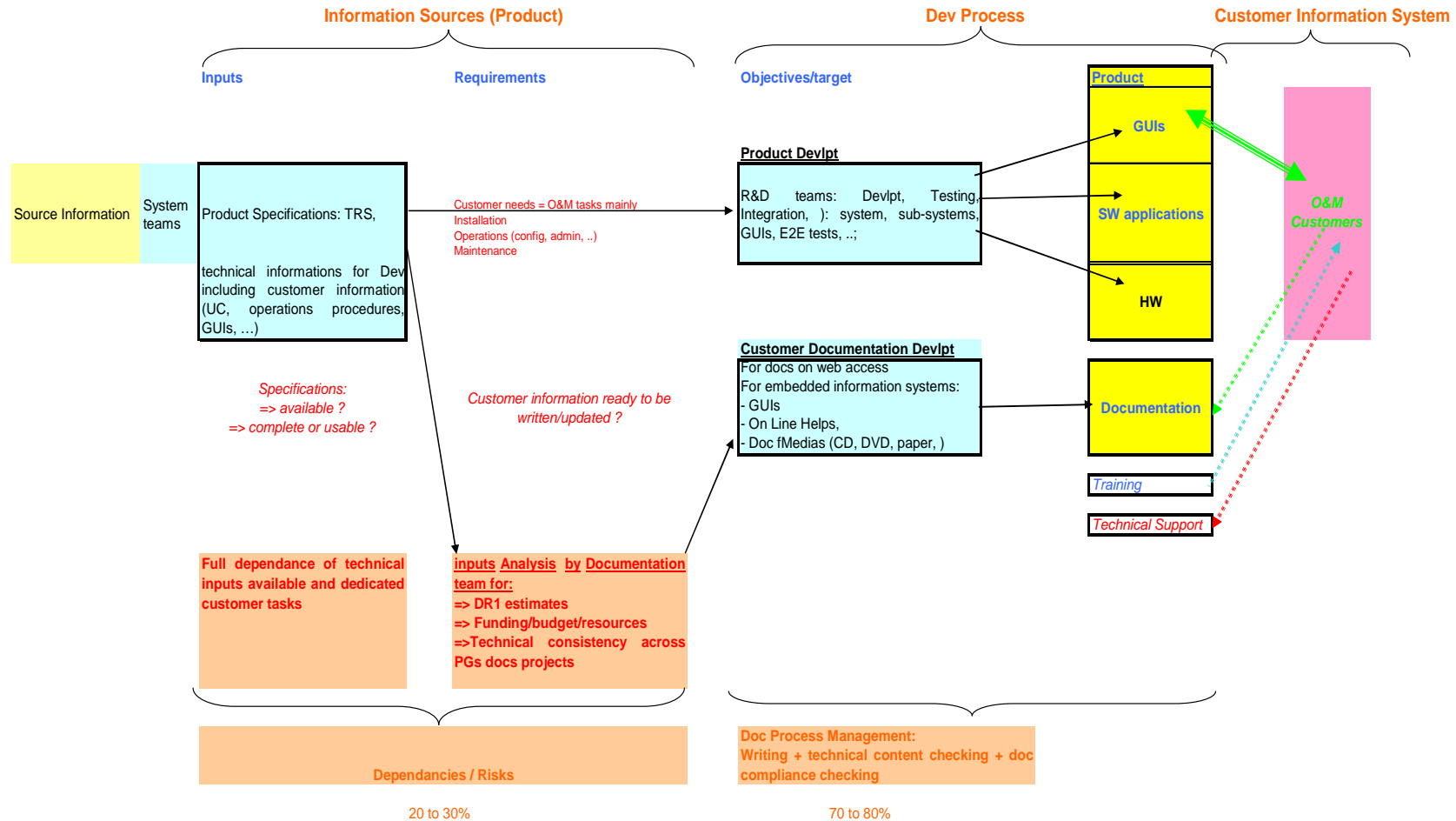
# Generating XML-DITA Procedure

```
Notepad++ - F:\Documents and Settings\Phil\My Documents\My Dev\Doc M_Lanque\Arkit_Sim\Exemples\Payment_NL\Module_Open_and_Handle_C...
File Edit Search View Format Language Settings Macro Run Plugins ?
Module_Open_and_Handle_Connect

5 <!-- Generated by ProcModeler - P.Larvet 03-26-2009 -->
6 <paramodule lang="EN">
7 <docstatus>
8 <refdoc codif="old">
9 <oldrefdoc>3BL00000000199ACBAPA</oldrefdoc><sheettype></sheettype>
10 </refdoc>
11 <titledoc>
12 <title1>7570_Open and Handle Connection with Peer_PROCEDURE</title1>
13 </titledoc>
14 <ednum>01</ednum>
15 <status statuscode="RL"/>
16 <lastmodifdate><year>30-0</year><month>6</month><day>08</day></lastmodifdate>
17
18 <descript>
19 <descriptitem>Use type: operation.</descriptitem>
20 <descriptitem>Customer: generic.</descriptitem>
21 <descriptitem>Comment: created for Diameter Agent domain.</descriptitem>
22 <descriptitem>Product/Platform release: from G3.1.</descriptitem>
23 <descriptitem>Reason for update: &mdash;.</descriptitem>
24 <descriptitem>Network solution(s): CORE.</descriptitem>
25 <descriptitem>Method(s): GUI.</descriptitem>
26 <descriptitem>External link(s): &mdash;.</descriptitem>
27 </descript>
28 <nature naturelibelle="procedure"/>
29 </docstatus>
30
31 <docontent>
32 <docbody>
33 <hfree>
34 <ie level1="Diameter Agent domain" level2="procedure" level3="Open and Handle Connection with Peer"/>
35 <ie level1="Peer" level2="Open and Handle Connection with"/>
36 <ie level1="Open and Handle Connection with" level2="Peer"/>
37 </hfree>
```

eXtensible Markup Language file nb char : 8266 Ln : 5 Col : 24 Sel : 0 Dos\Windows ANSI INS

# O&M Customer Information Management & Development Process





## Conclusion

---

- Within the context of **technical documentation**, we have presented an automated **process** for detecting the structure of **use cases** within **technical specifications** and for generating procedures in XML, in conformity with **DITA** standard.
- XML-DITA procedures are small, **reusable** documentation blocks, stored in a **Content Management System**
- XML-DITA procedures are directly usable to build **well-structured customer documentation**
- Beyond the old manual approach, this process helps technical document writers to build **reusable procedures** from the contents of **technical specifications**
- This process keeps **costs under control** to deliver **well-structured content to customers**