Structured Principle of the MSR-DCI

Abstract

This document describes the basic principels of the *MSR Document Control Instance* msr-dci and illustrates it by an case study.

It is intended to give an overview about this approach and it is not a reference for implementation issues.

The reader should be familiar with xml techniques like xsl and xschema.

Table of Contents

Introduction

Companies

MSR MEDOC [MSR-MEDOC]

Name Roles	Departement	Address	Contact
Hans-Jürgen Sonnleitner	Siemens V-DO Automotive AG		hans-juergen.sonnleitner@at.siemens.de
Herbert Klein	XI-Works	Olgastr.86 D-70180 Stuttgart	Phone +49 711 248398-10 FAX +49 711 248398-29 herbert.klein@xi-works.de www.xi-works.de

Version Information

Document Part	Editor			
	Company	Version	State	Remarks
3 RD 2003-07-11 For details refer to nr. 1, Page 26				
	MSR-MEDOC			

1 Introduction

With the recommondation of the xschema by the w3c in may 2001, xml instances now provide a lot more opportunities for semantic data checks.

Now it is possible to defines elements and attributes with certain data-types like *date*, *time*, *long*, *6 character string*, *list of values* etc. etc.

For detailed information look at see Standard [xschema2] p. 25.

This approach is a big step forward in particular for the exchange of engineering data. They consits often of non human readable information which have to be used in several tool chains and processes, which require strict data typing.

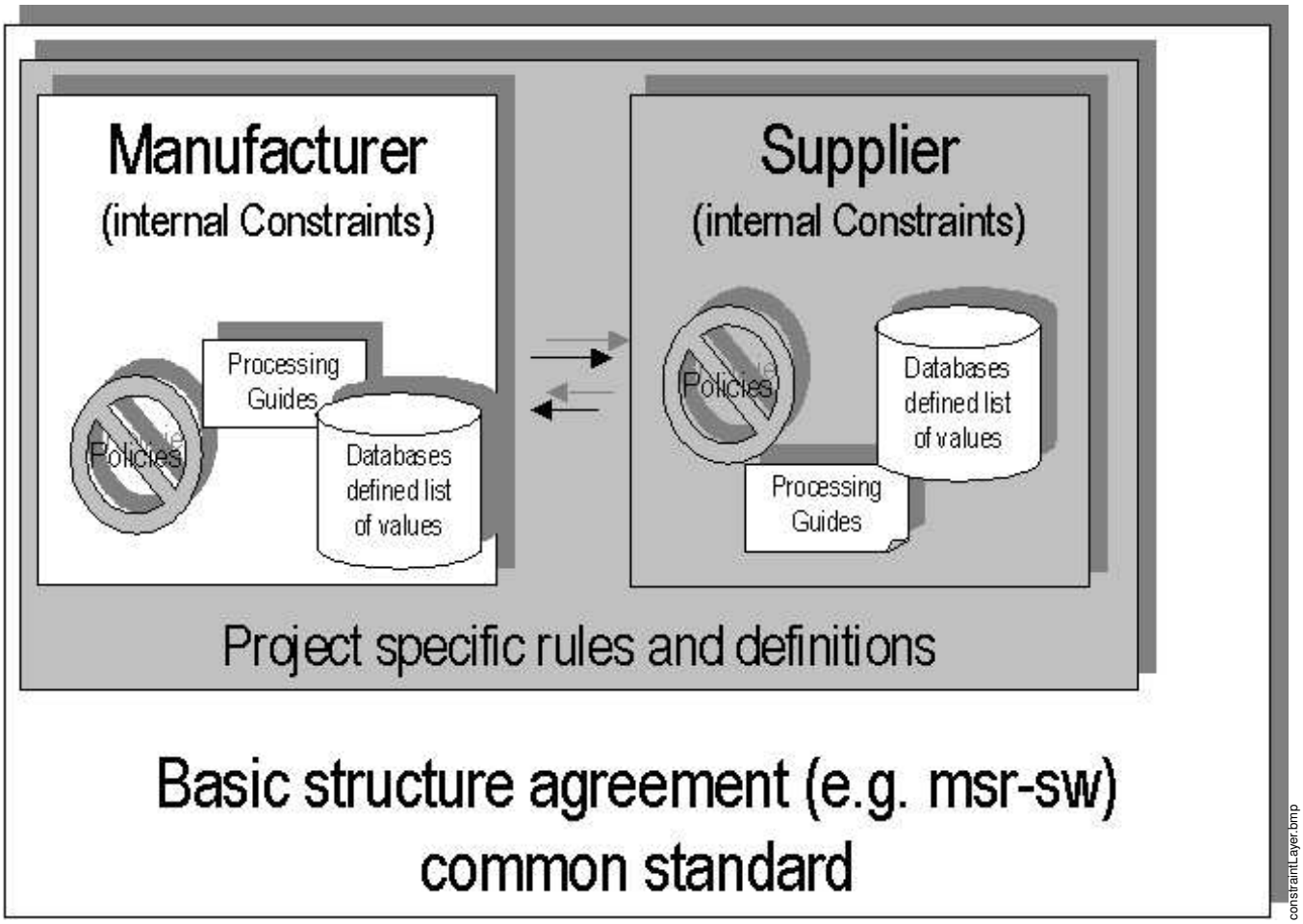
But there is still a lack of functionality!

Every constraints relate on a fixed strucuture. But what happen, if the structure itself (including list of values, attribute lists, mandatory elements...) changes by e.g. the maturity level of a information.

This will happen in the life-cycle of a document like a processing guide: If a document becames valid, a signature of a member of a specific group is mandatory, but not in the first initial draft phase.

Another usecase is e.g. in the manufacture/supplier relationship.

It is very useful, to have basic structure definitions (in a dtd or schema...) for exchanging information, but in addition to that every company may have different value lists for team-members, maturity levels due to internal policies or database related value lists for variable names.



To avoid now to develop and spread a bunch of DTD's for every company and for every process step the DCI approach allows the customization of a DTD to specific needs and support basic aggreements of the data-structures.

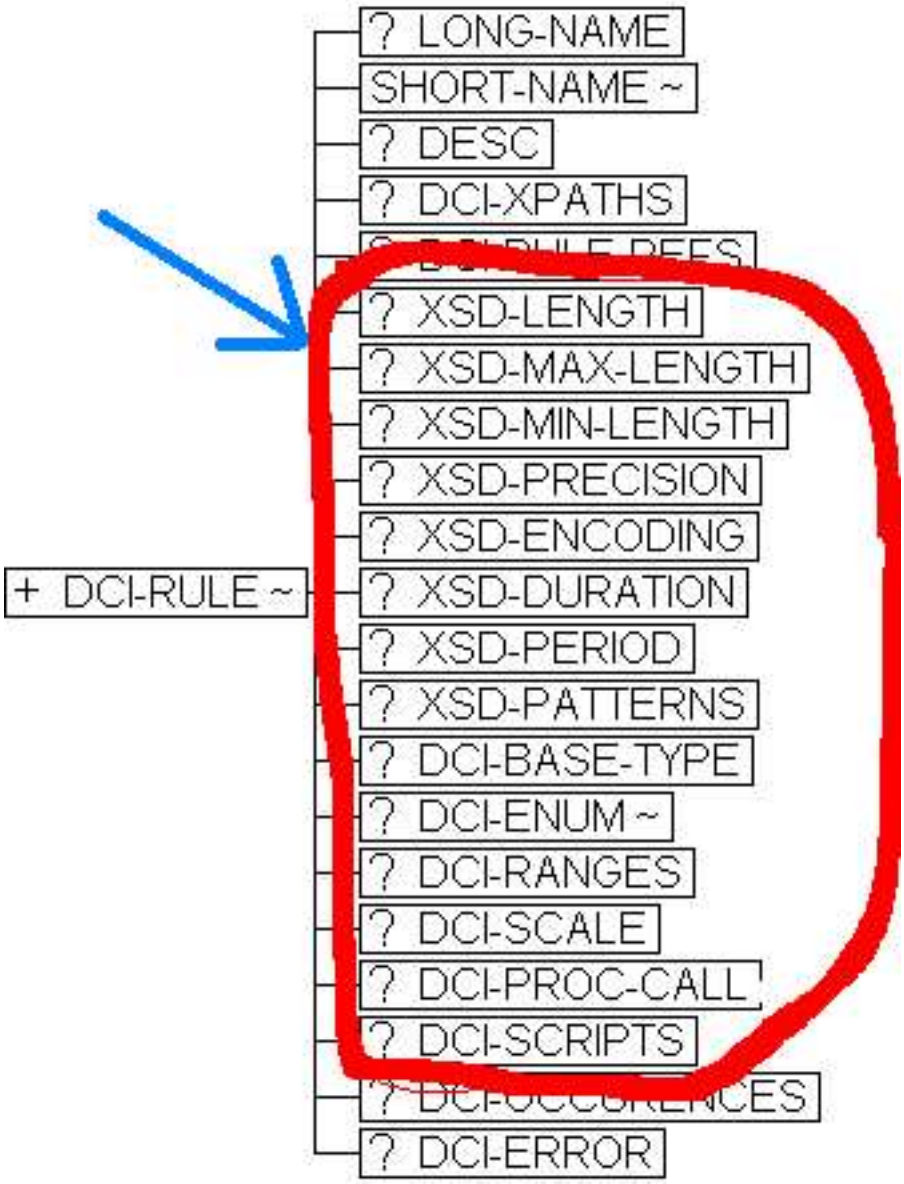
2 Basic Concept

The msr-dci is able to formulate constraints regarding the datatype and bind this rule to certain parts of the document by taken the content into account. Therefor we have two functinoalities - describing the datatype and formulate the binding.

2.1 Data type

This will be done by using the results of the W3C Schema group.

The possibility to express the data-type (list of values, regular expressions...) will by copied in the msr-structure and will be enlarged by a script functionality, which allows companies, to formulate advanced rules, which may need a connection to internal datasets.



dci-rule-data-type.bmp

11/07/2003 15:12:06 msrdci-sp-en.mmb

There is a intersection between the xschema activities of the W3C and the MSR dci approach. For several basic rules like length, minLength, enumeration ... we identified the same functionality in both. Due to this, the MSR uses the name of the W3C to ensure a common understanding regarding the datatype rules.

Elements, which have a correspondent in the W3C schema definition begins with xsd (xSchema-Datatype)

Elements, which are basic dci functionality begins with dci-

Basic MSR-Elements like short-name, long-name, desc have no prefix.

2.2 Structure dependency

Adressing the Elements

To combine the rules with the parts of the document/information set, a method must be choosen to formulate this dependency. E.g. to define that all **<short-name>** beneath the element **<sw-variable>** must not be longer than 31 characters we first have to adress these elements in a way that a program can select it:

Therefore the xpath-Recommondation of the w3c offers the right language: *//sw-variable/short-name* .

These expressions - the adressing of the elements - will be placed in **<DCI-XPATH>**

dci-xpath.bmp

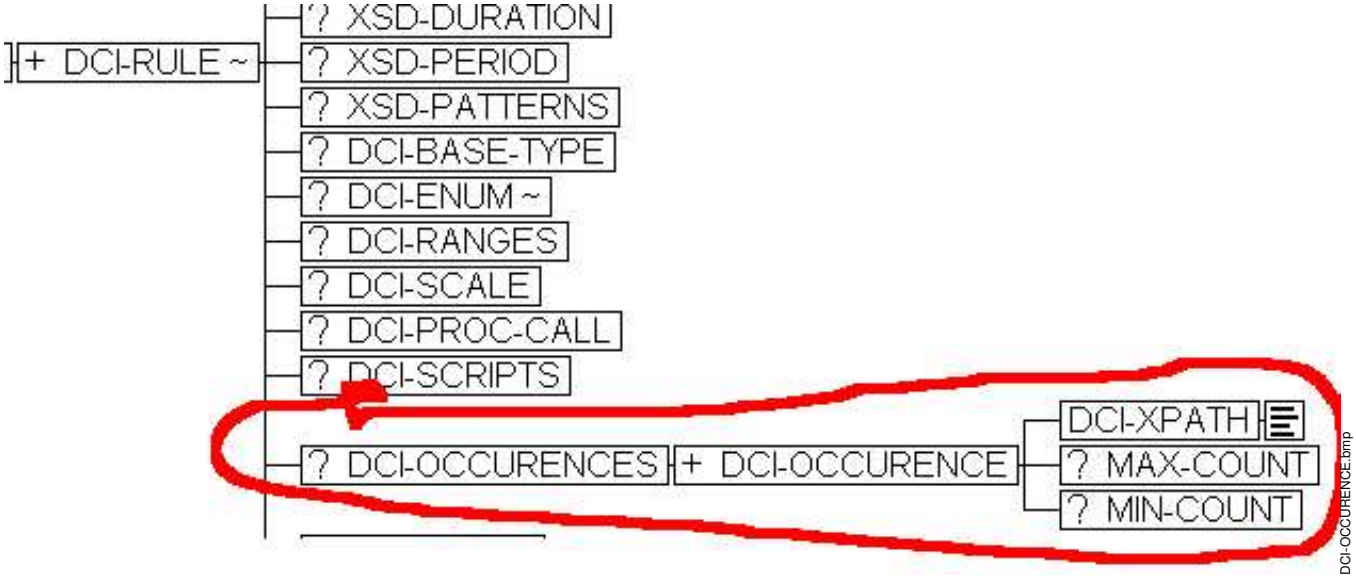
Define structure constraints

But up to now, it is still no possible, to express structural constraints like:

if the document has the maturity level "approved", the signature element is necessary.

Therfore the element **<DCI-OCCURENCE>** is able to formulate such an expression.

First, the xpath-Statment select the neccessary part of the document. Then, it will be looked, whether other elements have a defined occurence (in the example above - the signature element must occur 1 time.



3 Case study: E-mail system

To illustrate the possiblities of this approach, we will discuss it by an case study.
The e-mail example is a artificel one and has no real-life relation.

Imagine, that you are in charge to implement a mail-program.

Of cource, you will use for the realization XML technology and you will formulate the email with xml.

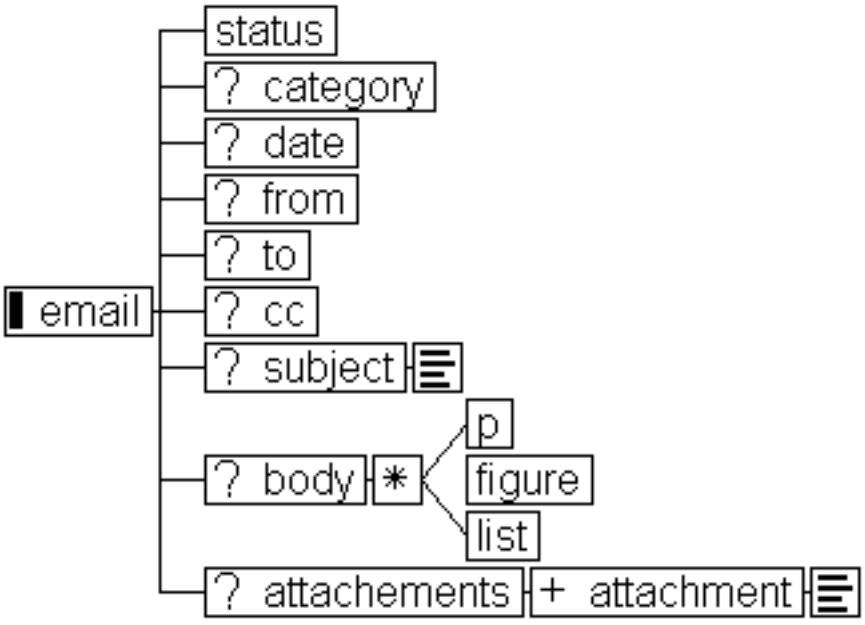
One of jobs we have to do, is to take serveral guidelines for the email exchange into account:

- Due to some restriction, the subject shouldn't be longer than 32 characters.
- Every mail must have a subject
- External mail will only be accepted if the sender is stored in a internal database table.

This case study now formulate these requirements with a DCI and show a possible implementation.

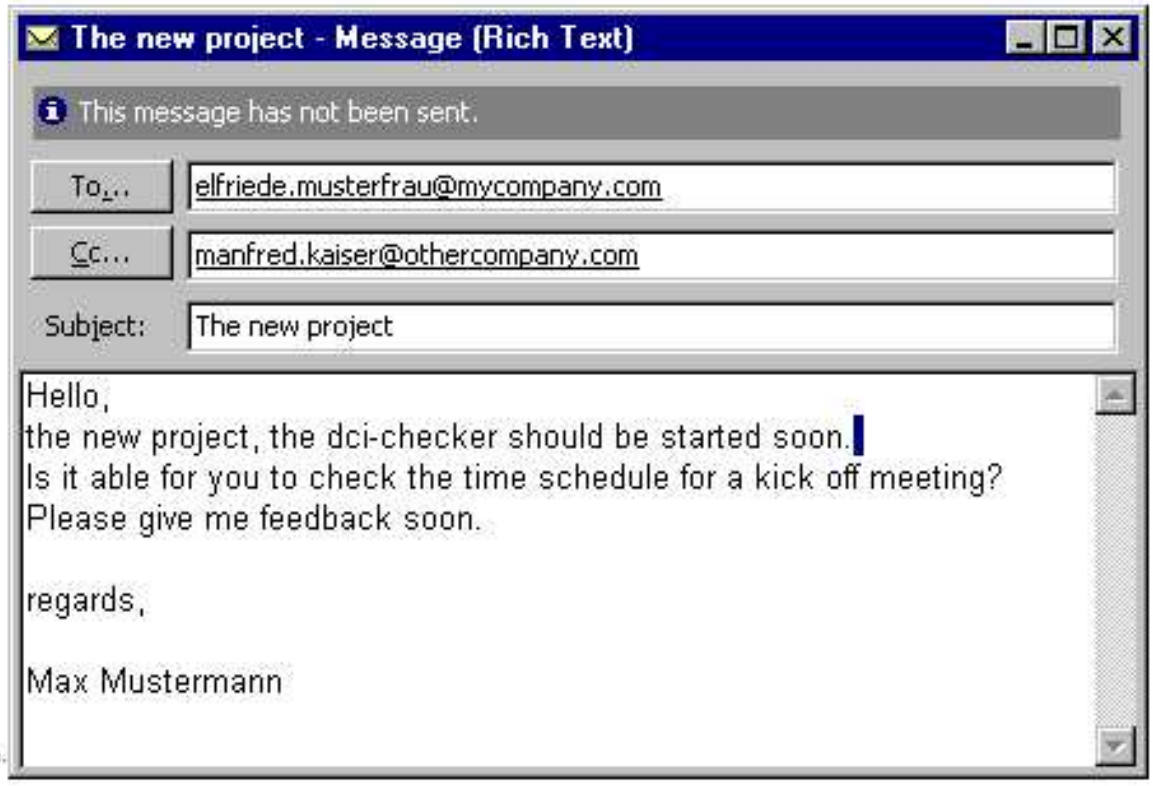
3.1 Preperation

First we have to design the DTD for the email-system and create a sample instance:



email-structure.bmp

email
status draft /status
from max.mustermann@mycompany.com /from
to elfriede.musterfrau@mycompany.com /to
cc manfred.kaiser@othercompany.com /cc
subject **The new project** /subject
body
p Hello, /p
p the new project, the dci-checker should be started soon. /p
p Is it able for you to check the time schedule for a kick off meeting?
/p
p Please give me feedback soon. /p
/body
/email



EMail-Dialog.bmp

3.2 Formulate the rules

The subject shouldn't be longer than 32 characters.

This constraint is easy to formulate - it is valid for all instances without any content related dependencies.

Therefore we use the element **<DCI-RULE/XSD-MAX-LENGTH>** and place the value 31 in.

```
LONG-NAME Length of Subject /LONG-NAME
  SHORT-NAME SubjectLenght /SHORT-NAME
  DESC The subject shouldn't be not longer than 31 characters. /DESC
  DCI-XPATHS
    DCI-XPATH //subject /DCI-XPATH
  /DCI-XPATHS
  XSD-MIN-LENGTH 1 /XSD-MIN-LENGTH
  XSD-MAX-LENGTH 31 /XSD-MAX-LENGTH
/DCI-RULE
```

rule-subjectlength.bmp

Every mail must have a subject

It is no problem to include this requirement into the DTD - mark the subject element as mandatory.

But we want to go one step further and offer the user more convienence - that means only when the user wants to send the mail, the subject is required, but not when he wants to store the mail as draft.

But let's start:

Define this rule from normal words up to dedicated xpath statements:

in "normal" wording	If the mail is ready to send, then a subject must be present.
with xml-words	If the element **<status>** has the value "ready to send", then the email must have a **<subject>** element.
with xpath-statements	If the xpath-expression *'//status[="ready to send']* returns true, then the xpath-expression *'//subject'* must return a non-zero value.

with the msr-dci

```
DCI-RULE
  LONG-NAME Every mail, which satisfies the internal policy must have a subject /LONG-NAME
  SHORT-NAME subject-occurence /SHORT-NAME
  DESC Before sending a mail, which is represented in the mail-status 'ready to send', the element
  subject must be filled out. /DESC
  DCI-XPATHS
    DCI-XPATH /email[status='ready to send'] /DCI-XPATH
  /DCI-XPATHS
  DCI-OCCURENCES
    DCI-OCCURENCE
      DCI-XPATH //subject /DCI-XPATH
      MIN-COUNT 1 /MIN-COUNT
    /DCI-OCCURENCE
  /DCI-OCCURENCES
/DCI-RULE
```

rule1.bmp

External mail will only be accepted if the sender is stored in a internal database table

This constraint is combined with external data-resources which are not defineable in DTD's or schemas. To cover such situation with the DCI, 2 elements allow the integration of thus external

resources - DCI-PROC-CALL and DCI-SCRIPT. Both have the possibility to execute program code and have with it the access to free defineable resources - as long as program code exist, which can access it.

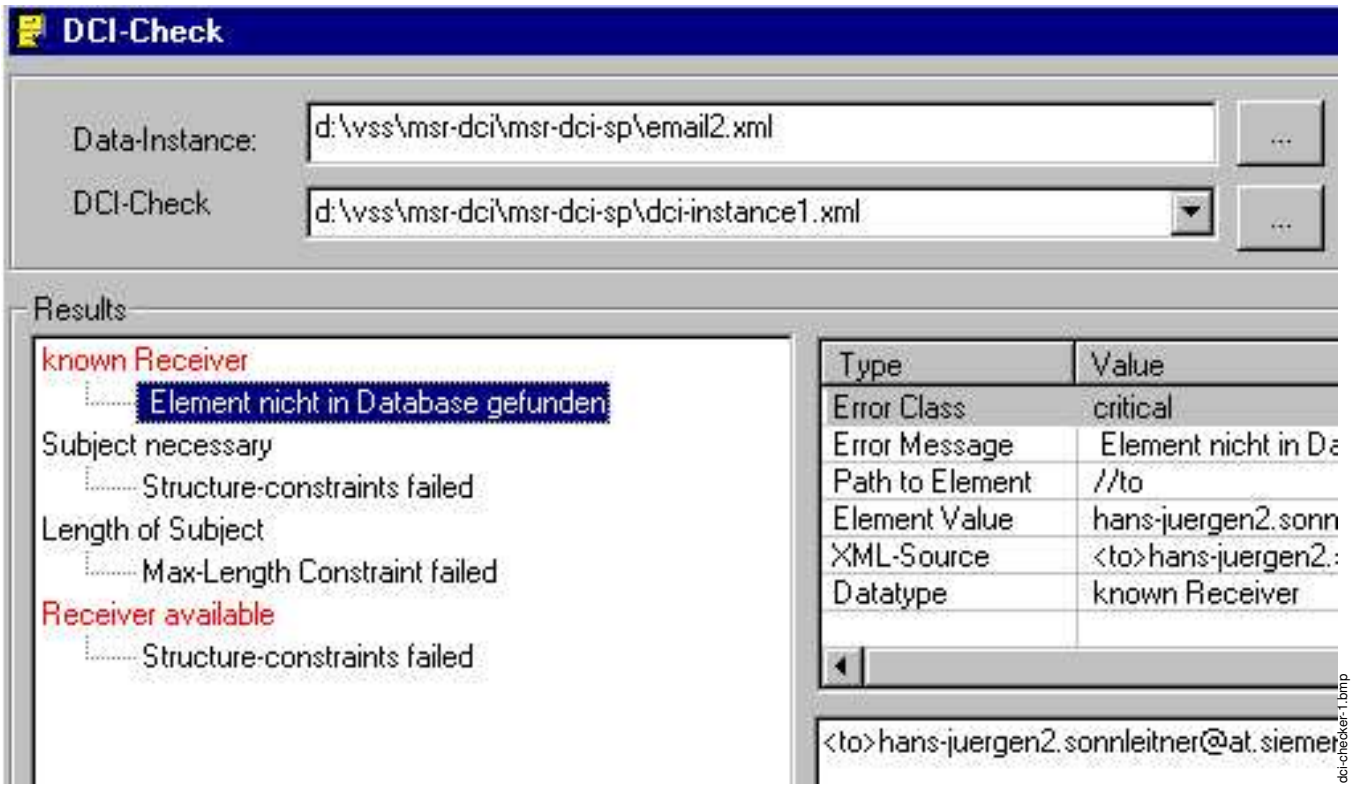
In the example we use a com-component, which can execute sql-statements.

```
DCI-RULE
  LONG-NAME known sender /LONG-NAME
  SHORT-NAME knownsender /SHORT-NAME
  DCI-XPATHS
    DCI-XPATH //from /DCI-XPATH
  /DCI-XPATHS
  DCI-PROC-CALL
    DCI-PROC-NAME DataChecks /DCI-PROC-NAME
    DCI-PROC-CLASS interface /DCI-PROC-CLASS
    DCI-PROC-ARG SQL-CALL /DCI-PROC-ARG
    DCI-PROC-ARG select * from tblemail where stremail like '[#value#]' /DCI-PROC-ARG
    DCI-PROC-ARG d:\msr\msr-dci\example\dci.mdb /DCI-PROC-ARG
  /DCI-PROC-CALL
  DCI-ERROR
    SHORT-NAME /SHORT-NAME
    DCI-ERR-MSG This receiver is not known! /DCI-ERR-MSG
    DCI-ERR-CLASS critical /DCI-ERR-CLASS
  /DCI-ERROR
/DCI-RULE
```

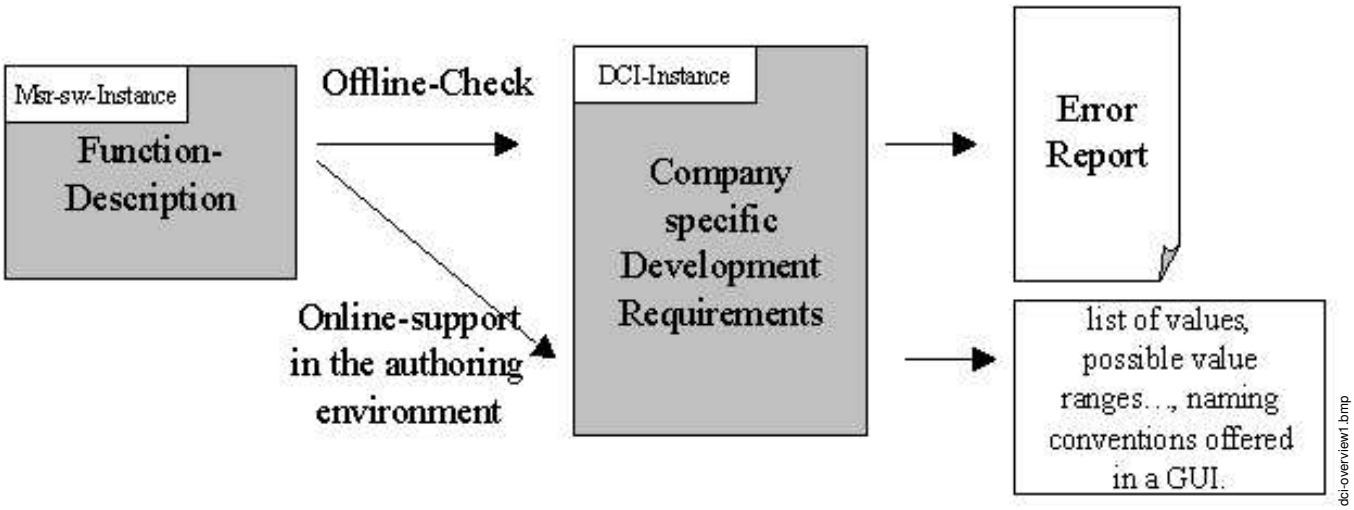
rule-known-sender.bmp

3.3 Running the constraints

With this defined rules, it is now able to check the instances with an dci-interpretor. As you see in the screenshot below, it is able to perform all checks and generate an error report.



3.4 Conclusion

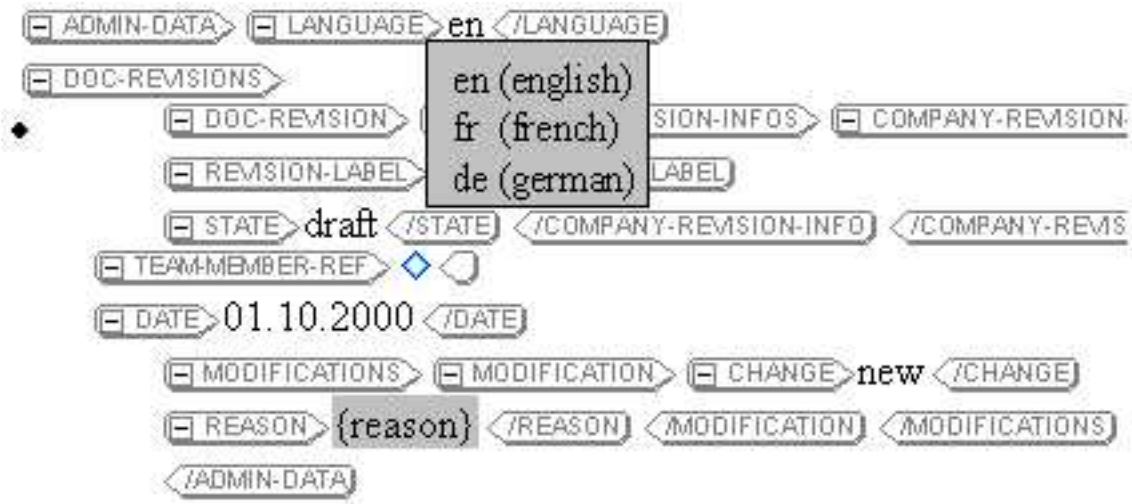


3.4.1 Author-Support - Online Processing

During the data acquisition the Author should be able to choose values for certain elements, for which the *msr-dci* instance provides the possible values.

It is also necessary to check element and attribute contents against the defined data-types in the dci.

E.g. a incorrect date or a wrong number format should be reported to the writer when the cursor leaves the element range or when a document check will be performed.



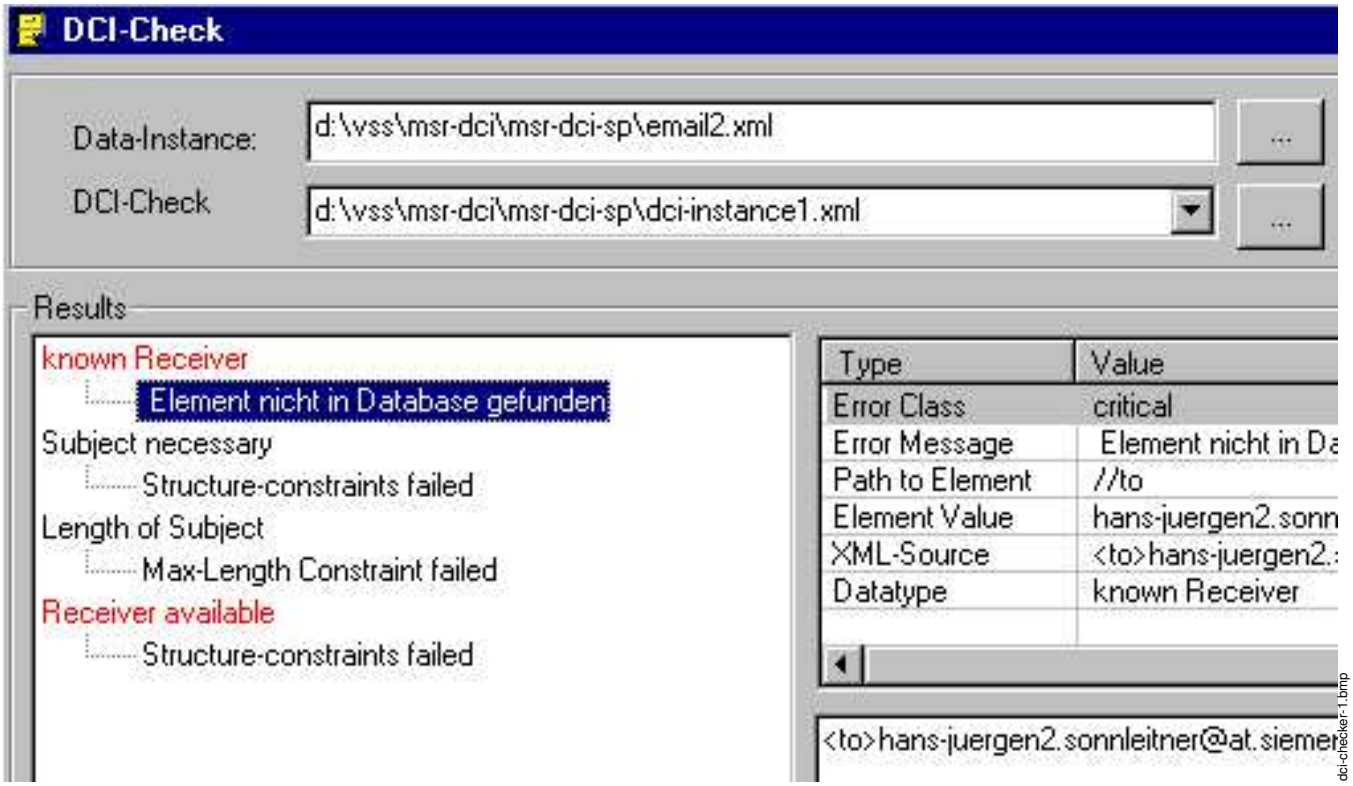
admin-data-1.bmp

3.4.2 Off line Processing

In the off-line Processing one or more instances can be completely checked again one or more dci instances. This can be useful for workflows or for the data exchange between companies or tool environments.

complete check of the instance.

for structural constraints.., error logs, status of development information....



dci-checker-1.bmp

11/07/2003 15:12:06 msrdci-sp-en.mmb

4 dci-architecture

The following chapters describes the basic structural elements of the DTD.

4.1 dci-specs/dci-spec

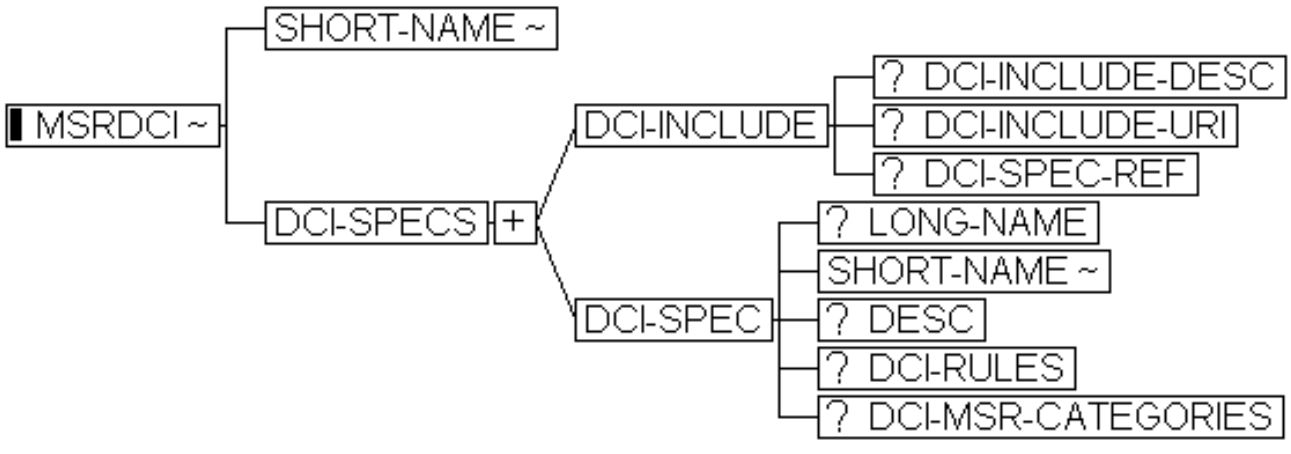
In a single DCI instance it is able to place several dci-spec structures as well as to include other DCI files.

This offer the possibility to cascade DCI-specifications.

E.g. some basic checks (e.g. the ASAM MCD basetypes) will be included. Company specific or development status related checks can place in this instance.

There are two possibilities to address the dci instance

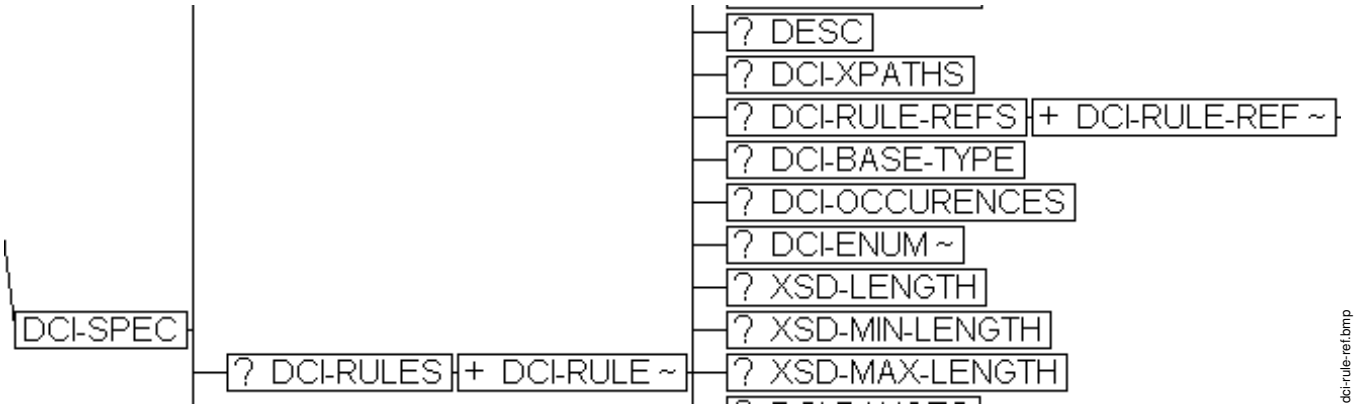
- with an unified resource locator - **<dci-include-uri>**
- with the **<dci-spec-ref>** element.



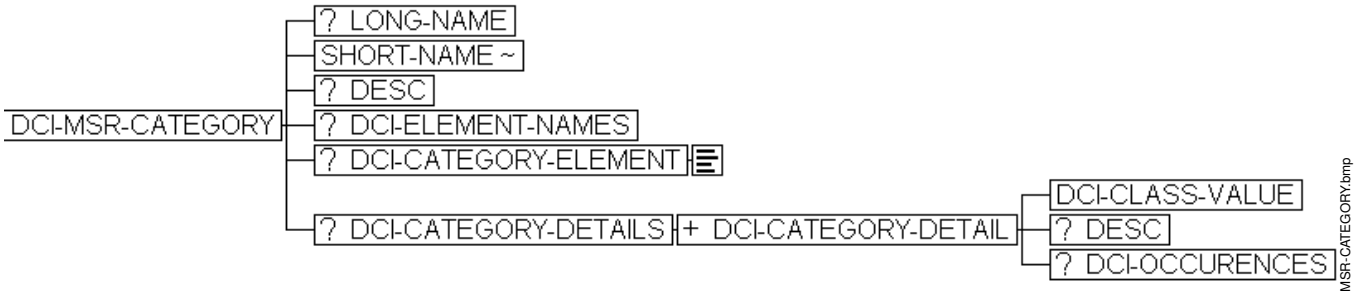
4.2 dci-rule-refs

This reference points to an certain **<dci-rule>**. It is up to the attribute **[impact]** , whether the scope of the rule set is extended or restricted.

With **<dci-rule-ref>** it is able to build hierachical designed data rules.



4.3 dci-msr-catetories



Here is a detailed example necessary

5 Defining the rules

5.1 Content related restrictions

The following chapters describe the possibilities how constraints regarding data in the instance can be formulated.

5.1.1 dci-base-type

This represents a build in base type. The check, wether a value is an integer or not, is not formulated in the datatype. For possible values we refer to some basic conventions.

A possible list of values can be found in : see External Document [xschema1] p. 25

5.1.2 dci-enum

A **<dci-enum>** represents a list of values.

Example:

The *ASAM* have a list for allowed base-types. A sw-instance can be checked , whether it use only allowed basetypes or not.

This element is corresponding to the W3C type enumeration

5.1.3 xsd-lenght

According to the W3C simple-type.

This Element specifies a exact length for a string

5.1.4 xsd-min-length/xsd-max-length

According to the W3C simple-types **<minLength>** and **<maxLength>**.

This Element specifies the minimal/maximal length for a string.

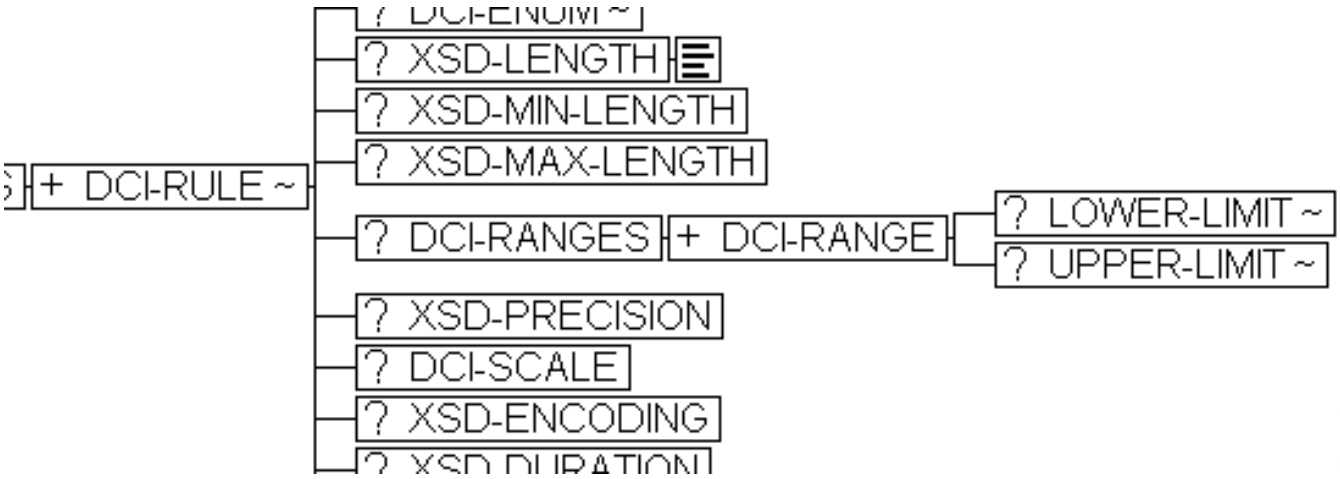
E.g. a non empty string has the:

```
<xsd-min-length>1</xsd-min-length>
```

5.1.5 dci-ranges

This element is corresponding to the xschema element**< maxinclusive>**, **<maxexclusive>**, **<mininclusive>**, **<minexclusive>**.

The min- and max- elements is expressed with the lower- and upper-limit element. The inclusive and exclusive information of the W3C is expressed with an attribute: **[intervaltype]**, which can be open or closed.



5.1.6 xsd-precision

<xsd-precision> is the maximum number of decimal digits in values of datatypes derived from decimal. The value of **<xsd-precision>** must be a positiveInteger.

5.1.7 dci-scale

< xsd-scale> is the maximum number of decimal digits in the fractional part of values of datatypes derived from decimal. The value of scale must be a nonNegativeInteger .

According to the W3C-Element **<scale>**

5.1.8 xsd-encoding

<xsd-encoding> is the encoded form of the lexical space of datatypes derived from binary. The value of encoding must be one of {hex, base64}.

According to the W3C-Element **<encoding>**

5.1.9 xsd-duration

<xsd-duration> is the duration of values for the datatype recurringDuration and datatypes derived from recurringDuration. The value of duration must be a timeDuration.

According to the W3C-Element **<duration>**

5.1.10 xsd-period

<xsd-period> is the frequency of recurrence for values for the datatype recurringDuration and datatypes derived from recurringDuration. The value of period must be timeDuration.

According to the W3C-Element **<period>**

5.1.11 xsd-pattern

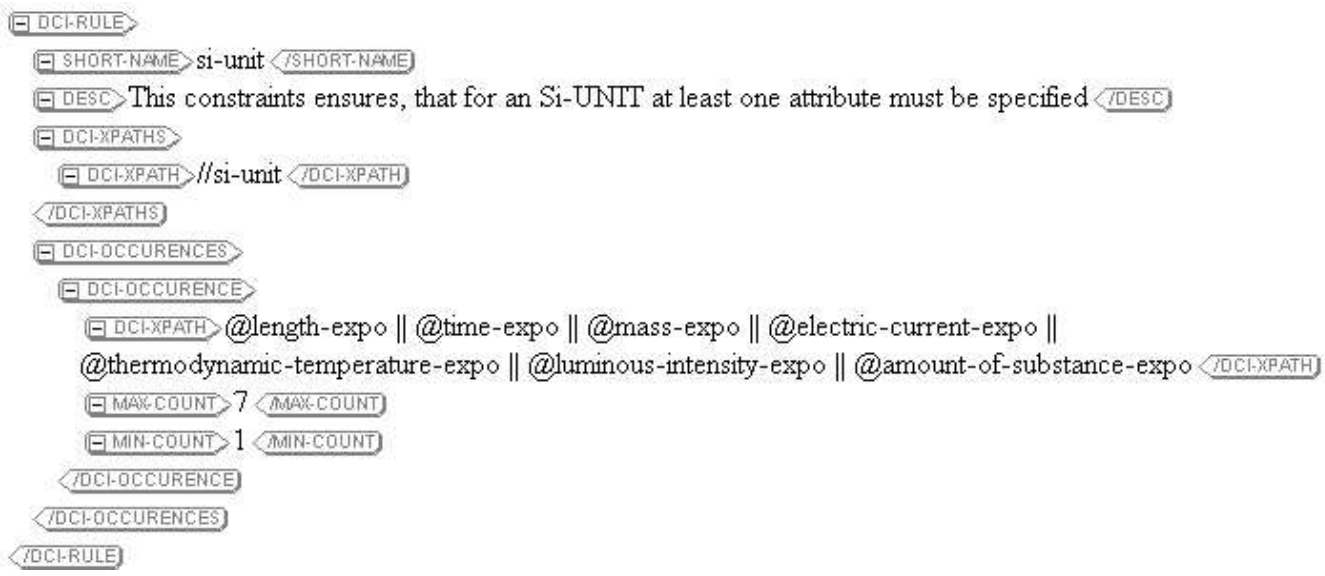
<xsd-pattern> is a constraint on the value space of a datatype which is achieved by constraining the lexical space to literals which match a specific pattern. The value of pattern must be a regular expression.

According to the W3C-Element **<pattern>**

5.2 structure-related restrictions

5.2.1 dci-occurence

This element defines the occurence of elements and branches in the DTD



OCCURENCE1.bmp

For the usage of this element the following guidelines must be considered

The xpath-expressions in the **<dci-occurence>** element is relative to the according **<dci-rule/dci-xpaths/dci-xpath>** element

5.3 advanced Restrictions

This elements offers advanced possibilities for processing the instance. They could be used when some constraints are not able to express with the simple datatypes elements.

5.3.1 DCI-PROC-CALL

This element allows to implement any thinkable constraint. E.g. the lookup of values in database-tables or the validation of element groups with specialized processors. For an example you can use the data, which are provided with this document.

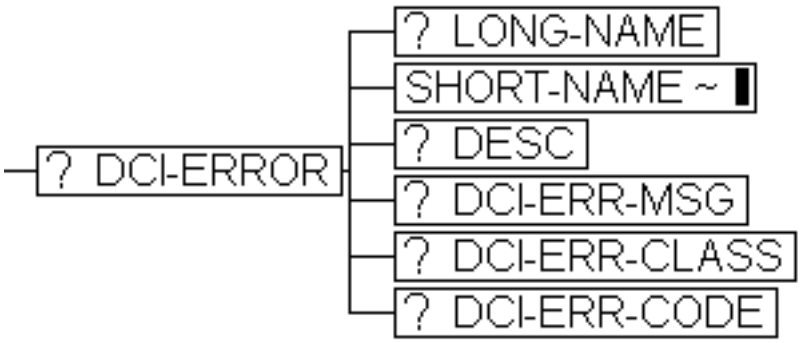
5.3.2 DCI-SCRIPTS

This element provide the place for the source of scripts which the dci-checker must be able to execute.

The access to the *dom* of the instance give big flexibility for further constraints.

6 Error Description

If there occure exceptions be processing the rules, a suitable error-message should be available.



DCI-ERROR.bmp

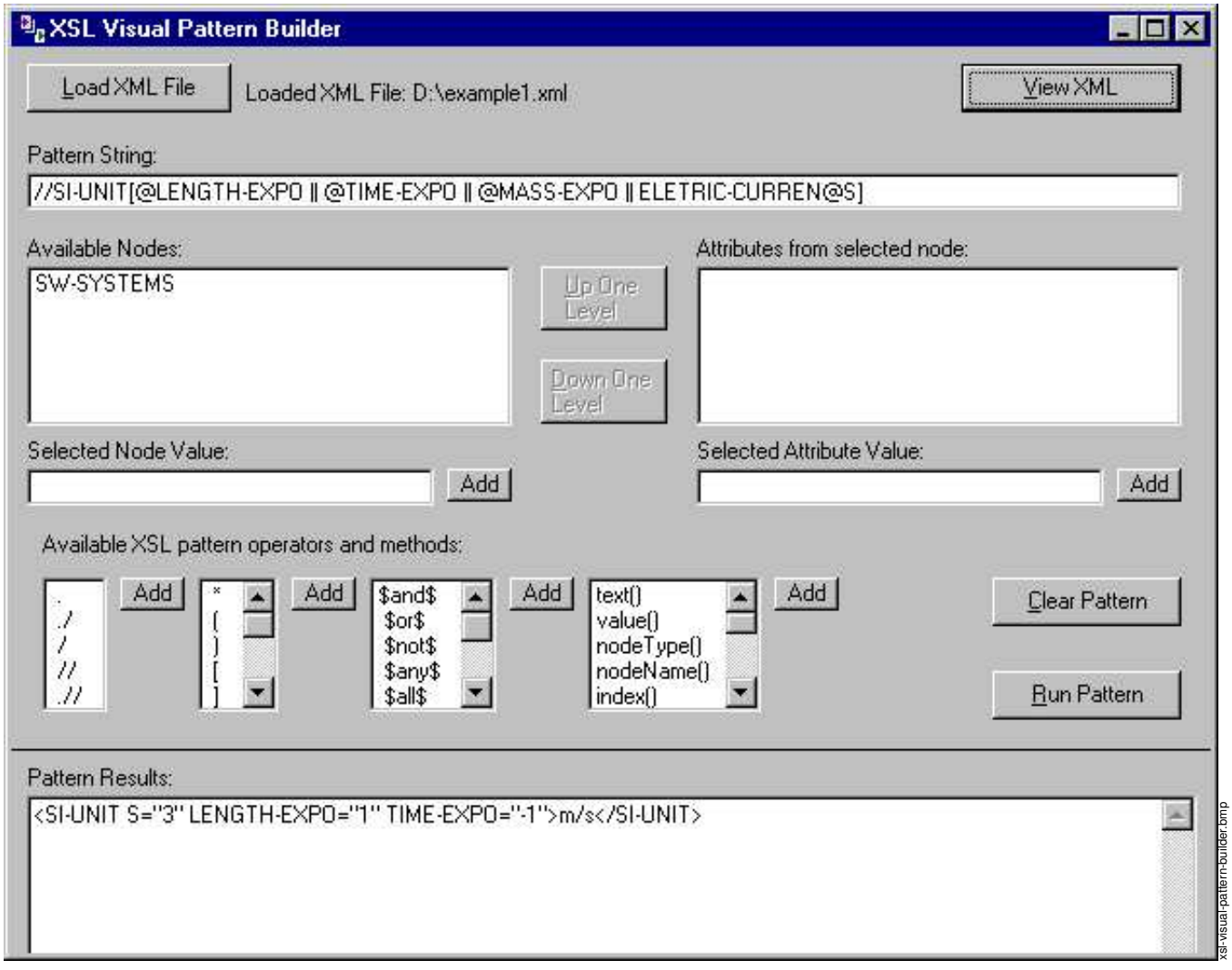
7 Addressing the elements

One of the main question is, how to adress the elements in the instance you want to check. Since November 1999, the answer can be a W3C recommendation - the XPATH standard.

These recommondation allows the authors of the dci to use a query language, which is tool independed and well described.

To use these query statements, some time is necessary, to understand the syntax.

A good tool for developing the xpath-statements is the visual pattern builder from Micrsoft. It is freeware, which is (including source) accessible on the Microsoft web site.



7.1 XPath Expressions

XPath is the basic language to query information from an XML-file. It offer the possibility to address all fragments inside an XML-instance for comparing it with the rules in the dci instance.

Find all cc elements within the current context:

```
./sw-unit
```

Note that this is equivalent to:

```
sw-unit
```

Find the root element (msrsw) of this document

```
/msrsw
```

Find all sw-unit elements anywhere within the current document:

```
//sw-unit
```

Find all sw-units where the value of the **<si-unit>** attribute **[mass-expo]** = 1

```
//sw-unit/si-unit[@mass-expo = "1"]
```

find all Si-units, where at least one of the attributes (lenght-expo to amount-of-substance-expo) are defined:

```
//si-unit[@length-expo || @time-expo || @mass-expo ||
  @electric-current-expo || @thermodynamic-temperature-expo ||
  @luminous-intensity-expo || @amount-of-substance-expo]
```

Find all **<short-name>** elements within an **<sw-unit>** element. Note that the **<sw-unit>** children of the current context are found, and then **<short-name>** children are found relative to the context of the **<sw-unit>** elements.

```
sw-unit/short-name
```

Find all **<short-name>** elements, one or more levels deep in the **<sw-data-dictionary-spec>** (arbitrary descendants):

```
sw-data-dictionary-spec//short-name
```

Note that this is different from the following pattern, which finds all short-name elements that are grandchildren of sw-data-dictionary elements:

```
sw-data-dictionary-spec/*/spec
```

Find all**< long-name>**s, one or more levels deep in the current context. Note that this situation is essentially the only one where the period notation is required:

```
.//long-name
```

Find all element children of **<sw-unit>** elements:

```
sw-unit/*
```

8 Additional documents and Resources

the msr-dci approach Microsoft XML-Parser Help-File [/ Standard: XML Path Language (XPath) Version 1.0 / State: Version 1.0 / Relevant Position: all] [/ Standard: XML Schema Part 1: Structures / State: Working Draft / Relevant Position: all] [/ Standard: XML Schema Part 2: Datatypes / State: Working Draft / Relevant Position: all] XSL und XPath Article about Schematron MSR-DCI-Checker MSR-DCI Editor

Documentadministration

Table : team members

Name	Company	
Hans-Jürgen Sonnleitner	MSR MEDOC	Department: Siemens VDO Automotive AG hans-juergen.sonnleitner@at.siemens.de
Herbert Klein	MSR MEDOC	Phone: +49 711 248398-10 Department: XI-Works herbert.klein@xi-works.de

Table : version overview

Version	Date	State
3	2003-07-11	RD
2	2003-06-16	RD

Table : modifications

Version	Change	Related to
3 : ,	Changed the some graphics Reason: -	Document

Table : modifications included

Date	Chapter	Change	Related to
Nr. 1, 2003-07-11	Gesamt	Changed the some graphics Reason: -	Document

References

Standards

Designation: **[xpath]:** XML Path Language (XPath) Version 1.0
State: Version 1.0
Relevant Position: all 25

Designation: **[xschema1]:** XML Schema Part 1: Structures
State: Working Draft
Relevant Position: all 25

Designation: **[xschema2]:** XML Schema Part 2: Datatypes
State: Working Draft
Relevant Position: all 25

External Documents

Designation: Article about Schematron
Date: 11/2000
Publisher: Heise-Verlag
Relevant Position: schematron 25

Designation: Microsoft XML-Parser Help-File
Relevant Position: Xpath 25

Designation: MSR-DCI Editor
Relevant Position: all 25

Designation: MSR-DCI-Checker
Relevant Position: all 25

Designation: the msr-dci approach
Relevant Position: all 25

Designation: XSL und XPath
Date: 2000
Publisher: addision-wesely
Relevant Position: all 25

Technical Terms

Code

Symbols

Control elements

0-9

D

L

T

Organisations

A

OTHER

Symbols

Products

M

SGML Attributes

I

M

SGML Elements

Symbols

D

E

M

P

S

X

Tools

D