Kermeta: Metamodeling Language

Breathe life into your metamodels

K.E.T. : Kermeta Emitter Template

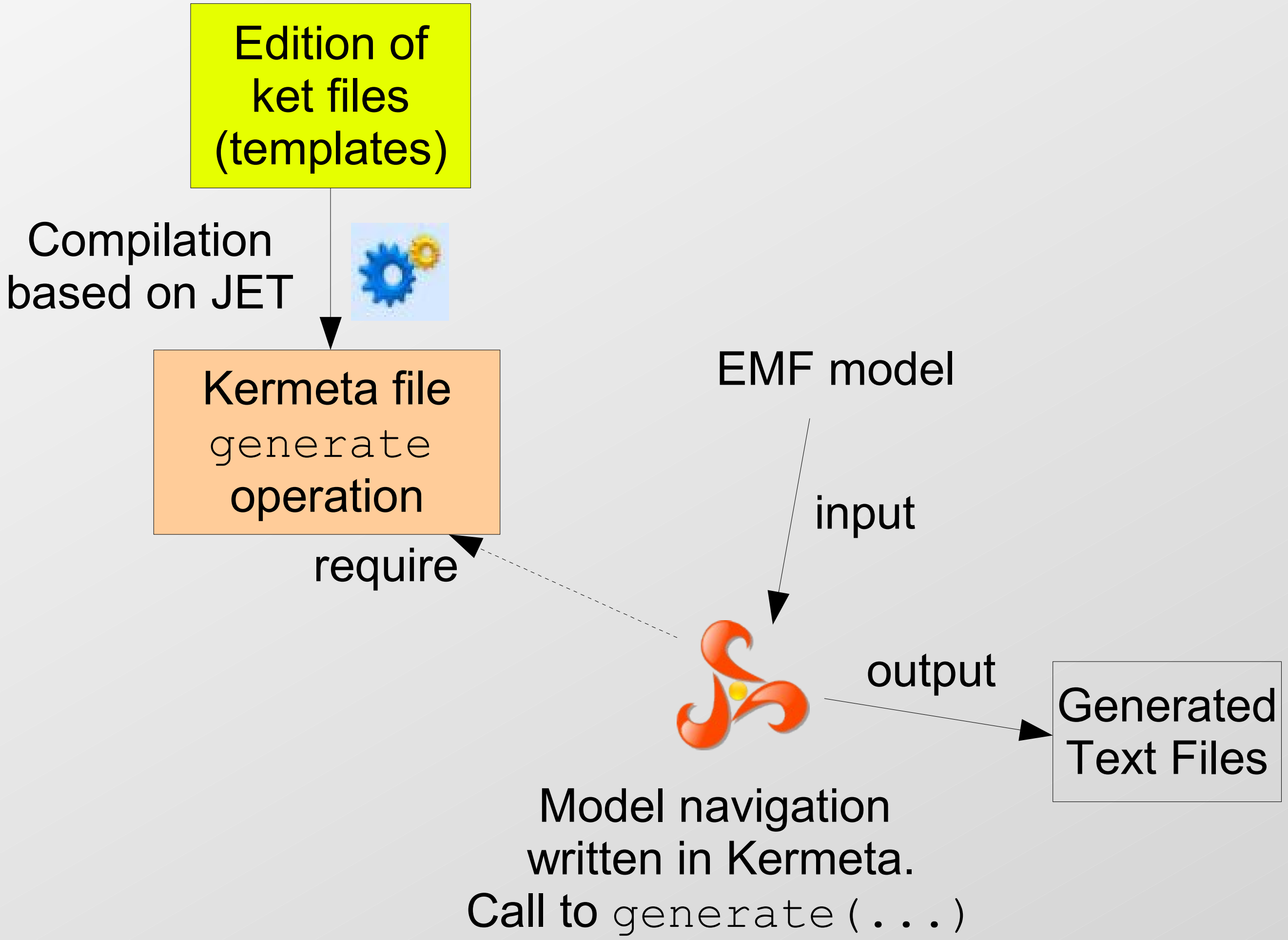
IRISA/INRIA – Triskell Team
Cyril Faucher, Brice Morin, Olivier Barais

KET: Kermeta Emitter Template

- A template engine to print text file
- Philosophy of JET, JSP
- Implementation based on JET
  (Java Emitter Templates from EMFT, a Eclipse subproject)
- Kermeta language is integrated in templates
- Unidirectionnal: Model to Text

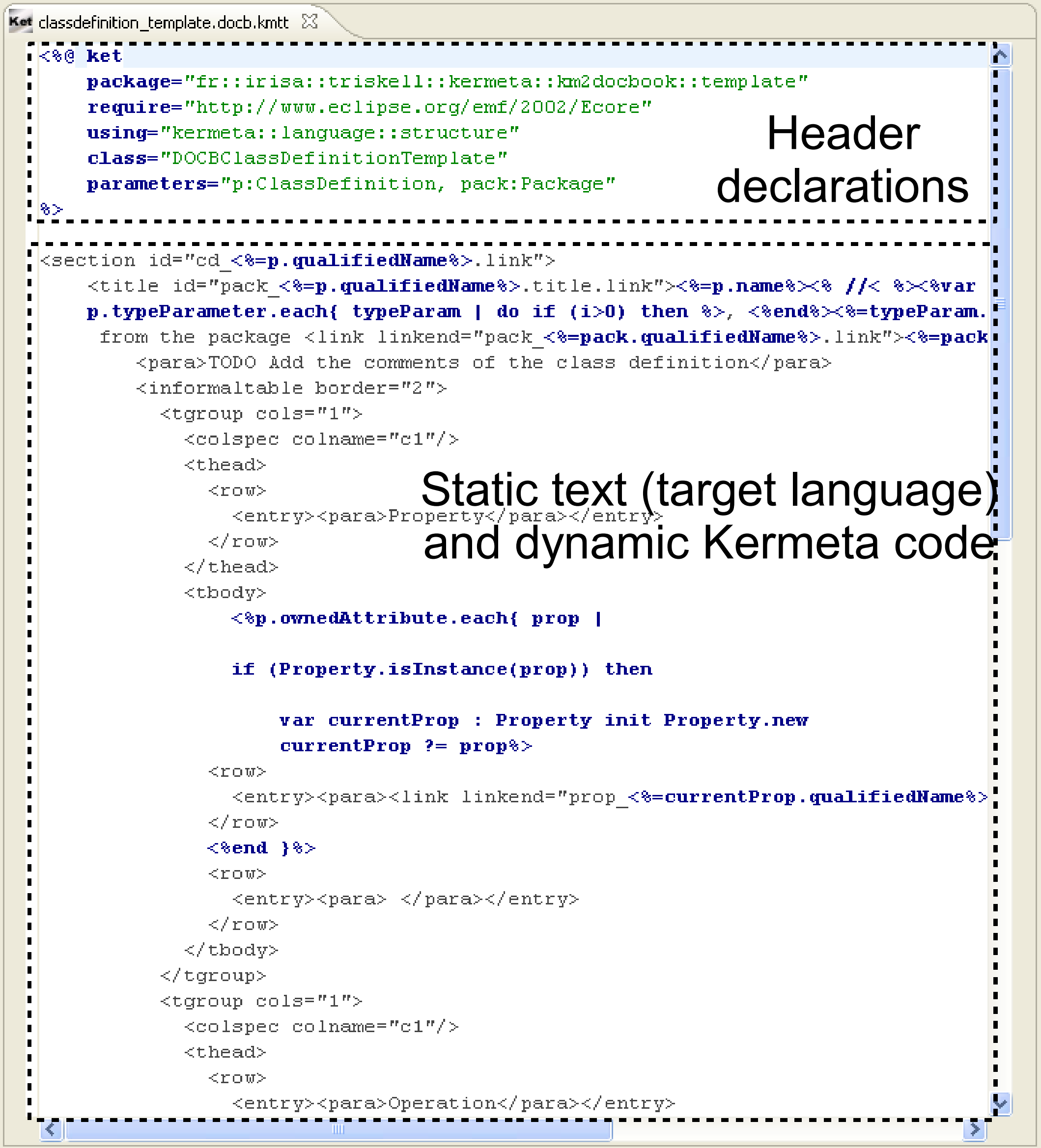
KET: Kermeta Emitter Template

- General process for using KET


Edition of ket files (templates)
Compilation based on JET
Kermeta file `generate` operation
EMF model
input
require
output
Generated Text Files
Model navigation written in Kermeta. Call to `generate(...)`

Kermeta

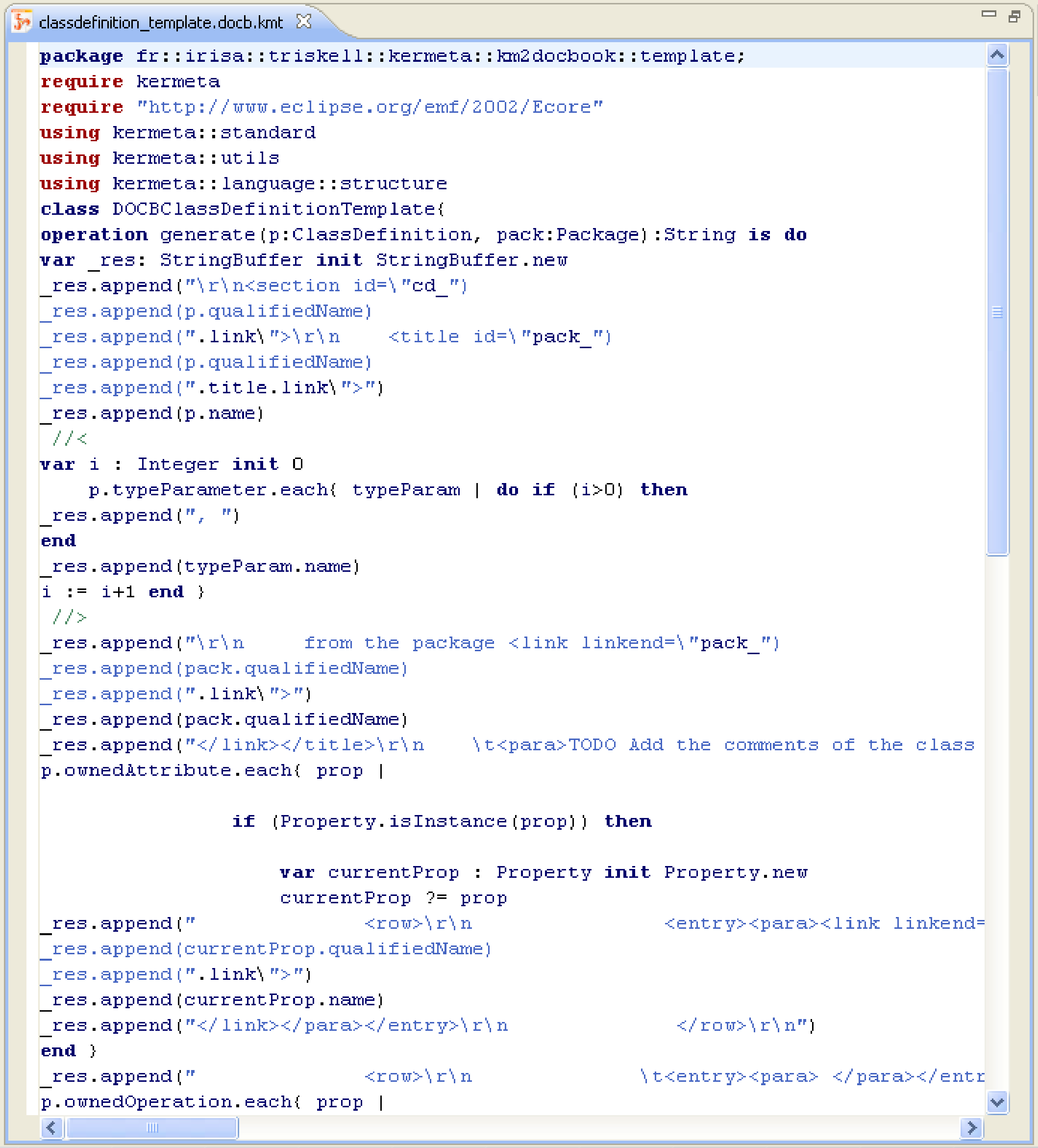
KET: Kermeta Emitter Template



```
<%@ ket
    package="fr::irisa::triskell::kermeta::km2docbook::template"
    require="http://www.eclipse.org/emf/2002/Ecore"
    using="kermeta::language::structure"
    class="DOCBClassDefinitionTemplate"
    parameters="p:ClassDefinition, pack:Package"
%>
<section id="cd_<%=p.qualifiedName%>.link">
    <title id="pack_<%=p.qualifiedName%>.title.link"><%=p.name%><% //< %><%var
    p.typeParameter.each{ typeParam | do if (i>0) then %>, <%end%><%=typeParam.
     from the package <link linkend="pack_<%=pack.qualifiedName%>.link"><%=pack
        <para>TODO Add the comments of the class definition</para>
        <informaltable border="2">
          <tgroup cols="1">
            <colspec colname="c1"/>
            <thead>
              <row>
                <entry><para>Property</para></entry>
              </row>
            </thead>
            <tbody>
                <%p.ownedAttribute.each{ prop |

                if (Property.isInstance(prop)) then

                    var currentProp : Property init Property.new
                    currentProp ?= prop%>
              <row>
                <entry><para><link linkend="prop_<%=currentProp.qualifiedName%>
              </row>
              <%end }%>
              <row>
                <entry><para> </para></entry>
              </row>
            </tbody>
          </tgroup>
          <tgroup cols="1">
            <colspec colname="c1"/>
            <thead>
              <row>
                <entry><para>Operation</para></entry>
```

Header declarations

Static text (target language) and dynamic Kermeta code



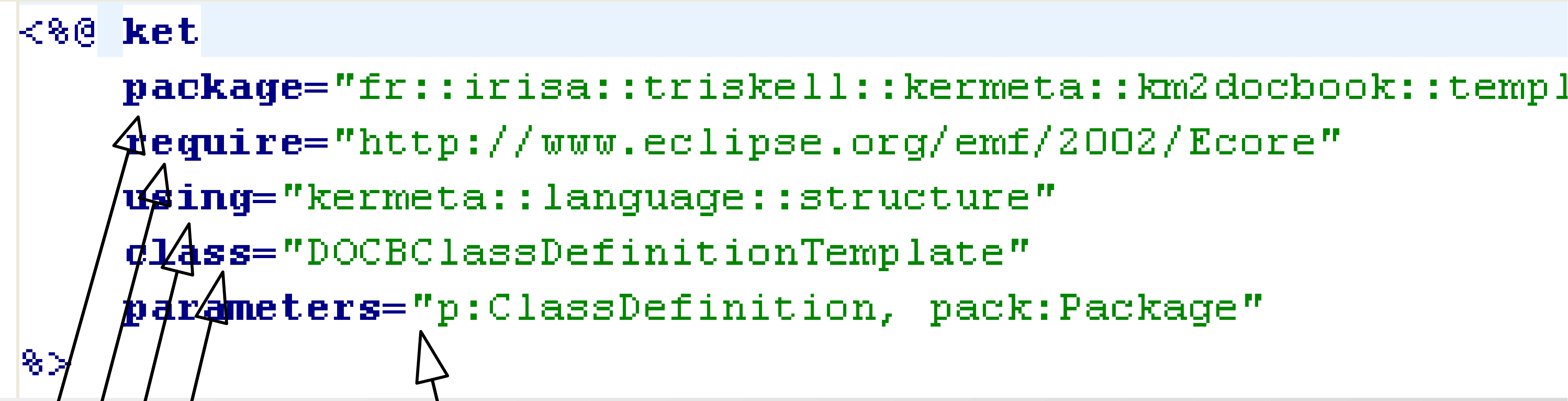
```
package fr::irisa::triskell::kermeta::km2docbook::template;
require kermeta
require "http://www.eclipse.org/emf/2002/Ecore"
using kermeta::standard
using kermeta::utils
using kermeta::language::structure
class DOCBClassDefinitionTemplate{
operation generate(p:ClassDefinition, pack:Package):String is do
var _res: StringBuffer init StringBuffer.new
_res.append("\r\n<section id=\"cd_")
_res.append(p.qualifiedName)
_res.append(".link\">\r\n    <title id=\"pack_")
_res.append(p.qualifiedName)
_res.append(".title.link\">")
_res.append(p.name)
 //<
var i : Integer init 0
    p.typeParameter.each{ typeParam | do if (i>0) then
_res.append(", ")
end
_res.append(typeParam.name)
i := i+1 end }
 //>
_res.append("\r\n     from the package <link linkend=\"pack_")
_res.append(pack.qualifiedName)
_res.append(".link\">")
_res.append(pack.qualifiedName)
_res.append("</link></title>\r\n    \t<para>TODO Add the comments of the class
p.ownedAttribute.each{ prop |

                if (Property.isInstance(prop)) then

                    var currentProp : Property init Property.new
                    currentProp ?= prop
_res.append("              <row>\r\n                <entry><para><link linkend=
_res.append(currentProp.qualifiedName)
_res.append(".link\">")
_res.append(currentProp.name)
_res.append("</link></para></entry>\r\n              </row>\r\n")
end }
_res.append("              <row>\r\n              \t<entry><para> </para></entr
p.ownedOperation.each{ prop |
```

ket template

Compiled *ket* into *kmt* file

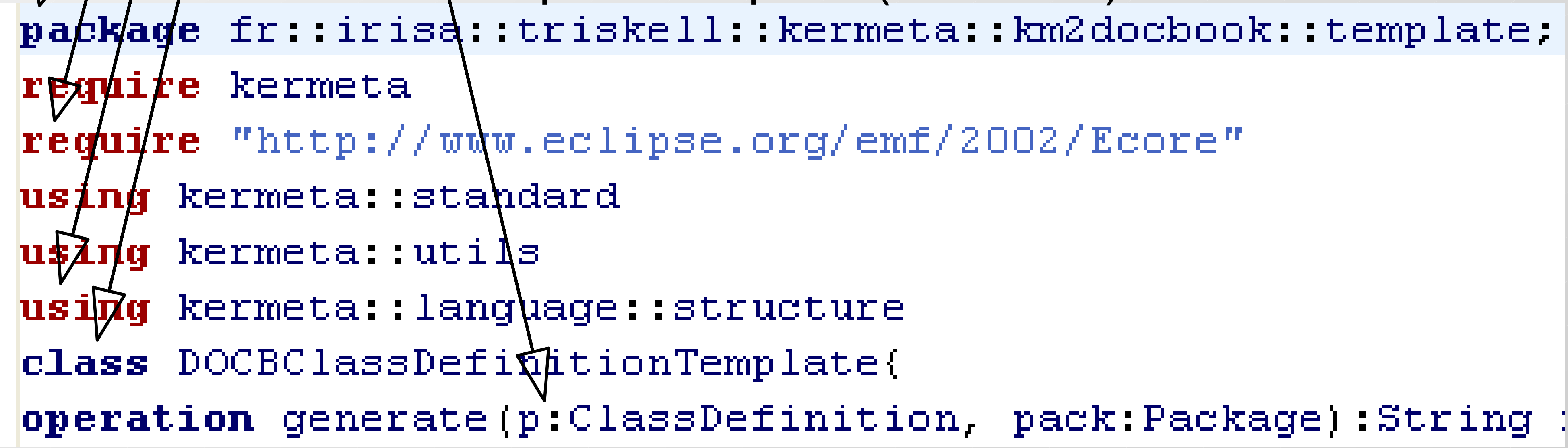
Mapping KET - KMT

- Header of a Ket file:

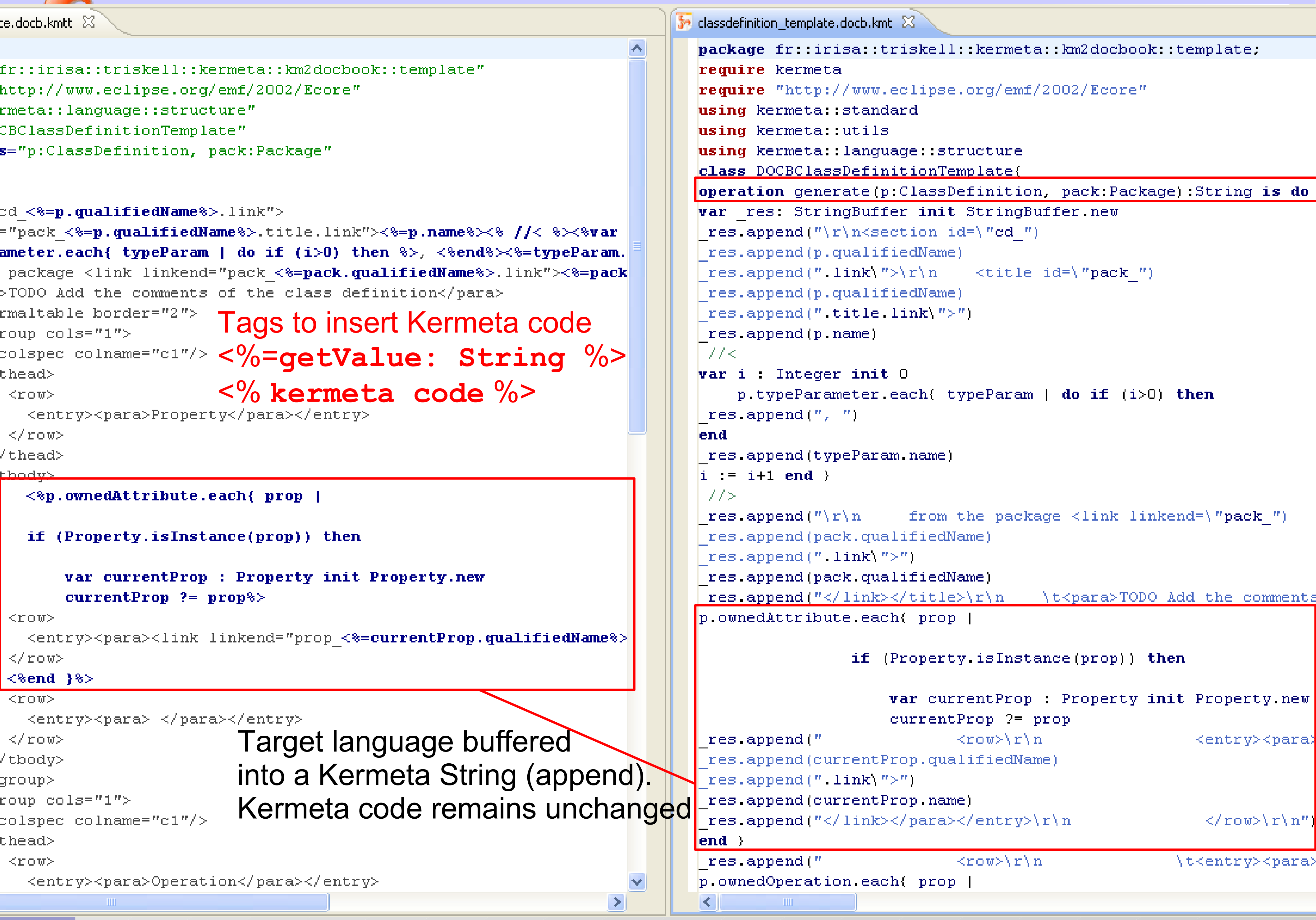


```
<%@ ket
    package="fr::irisa::triskell::kermeta::km2docbook::templ
    require="http://www.eclipse.org/emf/2002/Ecore"
    using="kermeta::language::structure"
    class="DOCBClassDefinitionTemplate"
    parameters="p:ClassDefinition, pack:Package"
%>
```

- Header of the compiled template (=> in kmt):



```
package fr::irisa::triskell::kermeta::km2docbook::template;
require kermeta
require "http://www.eclipse.org/emf/2002/Ecore"
using kermeta::standard
using kermeta::utils
using kermeta::language::structure
class DOCBClassDefinitionTemplate{
operation generate(p:ClassDefinition, pack:Package):String :
```



```
fr::irisa::triskell::kermeta::km2docbook::template"
http://www.eclipse.org/emf/2002/Ecore"
rmeta::language::structure"
CBClassDefinitionTemplate"
s="p:ClassDefinition, pack:Package"


cd_<%=p.qualifiedName%>.link">
="pack_<%=p.qualifiedName%>.title.link"><%=p.name%><% //< %><%var
ameter.each{ typeParam | do if (i>0) then %>, <%end%><%=typeParam.
 package <link linkend="pack_<%=pack.qualifiedName%>.link"><%=pack
>TODO Add the comments of the class definition</para>
rmaltable border="2">
roup cols="1">
colspec colname="c1"/>
thead>
 <row>
   <entry><para>Property</para></entry>
 </row>
/thead>
tbody>
   <%p.ownedAttribute.each{ prop |

   if (Property.isInstance(prop)) then

       var currentProp : Property init Property.new
       currentProp ?= prop%>
 <row>
   <entry><para><link linkend="prop_<%=currentProp.qualifiedName%>
 </row>
 <%end }%>
 <row>
   <entry><para> </para></entry>
 </row>
/tbody>
group>
roup cols="1">
colspec colname="c1"/>
thead>
 <row>
   <entry><para>Operation</para></entry>
```

```
package fr::irisa::triskell::kermeta::km2docbook::template;
require kermeta
require "http://www.eclipse.org/emf/2002/Ecore"
using kermeta::standard
using kermeta::utils
using kermeta::language::structure
class DOCBClassDefinitionTemplate{
operation generate(p:ClassDefinition, pack:Package):String is do
var _res: StringBuffer init StringBuffer.new
_res.append("\r\n<section id=\"cd_")
_res.append(p.qualifiedName)
_res.append(".link\">\r\n     <title id=\"pack_")
_res.append(p.qualifiedName)
_res.append(".title.link\">")
_res.append(p.name)
 //<
var i : Integer init 0
    p.typeParameter.each( typeParam | do if (i>0) then
_res.append(", ")
end
_res.append(typeParam.name)
i := i+1 end )
 //>
_res.append("\r\n      from the package <link linkend=\"pack_")
_res.append(pack.qualifiedName)
_res.append(".link\">")
_res.append(pack.qualifiedName)
_res.append("</link></title>\r\n     \t<para>TODO Add the comments
p.ownedAttribute.each( prop |

                if (Property.isInstance(prop)) then

                    var currentProp : Property init Property.new
                    currentProp ?= prop
_res.append("             <row>\r\n                 <entry><para
_res.append(currentProp.qualifiedName)
_res.append(".link\">")
_res.append(currentProp.name)
_res.append("</link></para></entry>\r\n             </row>\r\n")
end )
_res.append("             <row>\r\n              \t<entry><para>
p.ownedOperation.each( prop |
```

Tags to insert Kermeta code

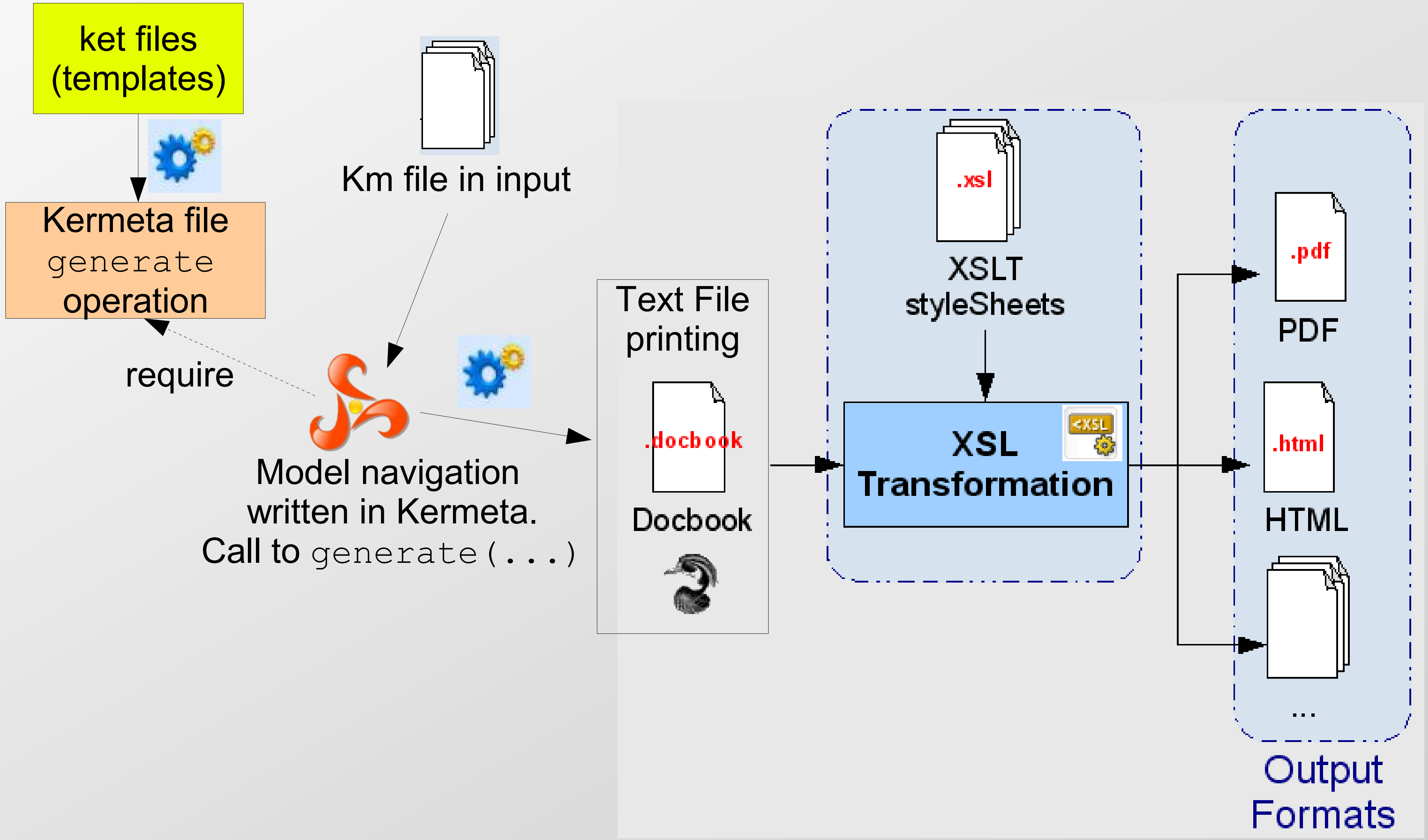
<%=getValue: String %>

<% kermeta code %>

Target language buffered into a Kermeta String (append).

Kermeta code remains unchanged

Documenting a Kermeta metamodel


ket files
(templates)
Kermeta file
generate
operation
Km file in input
require
Model navigation
written in Kermeta.
Call to generate(...)
Text File
printing
.docbook
Docbook
.xsl
XSLT
styleSheets
XSL
Transformation
<XSL
.pdf
PDF
.html
HTML
...
Output
Formats