My view:

Concept	**Concept representation in UML**	**... in eCore/EMF**
Invariant constraint	UML constraint with OCL opaque expression (OCL invariant keyword)	constraint annotation + ValidationDelegate
Initial value	Default value (could but needn't be an OCL opaque expression)	Nothing yet, see bug #405065
Derivation rule	UML constraint with OCL opaque expression on the **property** (OCL **derivation** keyword) **[1]**	derivation annotation + SettingDelegate **[2]**

In this view, what is currently lacking in MDT/OCL, UML2 and papyrus is

- a way to specify [1] on the UML/papyrus side of things, where the current xtext/OCL integration supposes that each constraint is an invariant constraint. The possibility of distinguishing between an OCL invariant constraint and an OCL derivation should be added.
- a way to transform [1] into [2]

This view seems to be also inline with for instance figure 5.5 in the book of Warmer and Kleppe.

Your view (differences highlighted in **bold font**):

Concept	**Concept representation in UML**	**... in eCore/EMF**
Invariant constraint	UML constraint with OCL opaque expression (OCL invariant keyword)	constraint annotation + ValidationDelegate
Initial value	Default value (could but needn't be an OCL opaque expression)	Nothing yet, see bug #405065
Derivation rule	UML constraint with OCL opaque expression on the **class** (OCL **invariant** keyword) **[1]**	derivation annotation + SettingDelegate **[2]**

In this case:

- You specify the UML constraint for the derivation rule in the context of the class instead of on the property
- You don't use the "derivation" part of OCL in the UML tooling (and hence you get a somewhat a-symmetry between UML and eCore+OCL tooling in which the derivation is available)
- You don't need to do anything to get this working with the current tools
- In order to transform a UML model [1] into an eCore model with correct annotations [2] you need to do more interpretation/work than in the first version (i.e. you need to couple the "derived" information on the properties with the invariant constraints on the class level. [*]