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	`isDerived = true` + Default Value with OCL opaque expression for the **property** **[1]**	derivation annotation + SettingDelegate **[2]**

In this case:

- You use the default value for specifying the OCL derivation rule as an initial value doesn't make sense in the case of a derived property.
- You don't need to do anything to get this working with the current tools, although you need to know some intrinsics of papyrus until `https://bugs.eclipse.org/bugs/show_bug.cgi?id=399249#c8` is solved.
- Bug #404876 needs to be solved in order to be able to transfer this into ecore