Let statement

1) Suppose we want to model the criterion: activity intolerance due to (low back pain or back-related leg symptoms) not improving over 4 weeks

```
Last activity_intolerance where (relationship.name == caused_by and
(relationship.patient_data ==  (back_pain where (body_site_cd == low_back) ) or
relationship.patient_data ==  (symptom where (body_site_cd == leg) and
relationship.name == related and relationship.patient_data == back_problem))) and
critical_time.low < now - 4 weeks). severity <
Latest activity_intolerance where (relationship.name == caused_by and
(relationship.patient_data ==  low_back_pain or relationship.patient_data ==  (symptom
where (body_site_cd == leg and relationship.name == related and
relationship.patient_data == back_problem)))).severity
```

But this is very hard to comprehend. Instead, we can break it up into 4 statements:

```
Let activity_intolerance where (relationship.name == caused_by and
(relationship.patient_data ==  (back_pain where (body_site_cd == low_back) ) or
relationship.patient_data ==  (symptom where (body_site_cd == leg and
relationship.name == related and relationship.patient_data == back_problem))))
be AI;

Let (Last AI where critical_time.low < now  - 4 weeks) be L1;

Let (Latest AI) be L2;

L1.severity < L2.severity
```

2) Where should the Let statement be defined and how should it be used?
The Let statement can be thought of as a macro from the C programming language. The macro has a constant_name that is expanded into a longer expression every time the constant_name is encountered.

I propose that macro definitions should be defined at the level of a guideline and can be used by that guideline, (not by its subguidelines, and not by guidelines that are using the guideline as a subguideline).

The guideline can have the following attribute:

```
        macro_definitions: Macro_Definition
```

```
Macro_Definition
        constant_name: String             // e.g., L2
        expression: Expression            // e.g., Latest AI
```

An alternative approach is to create a subguideline for the decision step. In this subguideline we can use *assignment actions* to assign the results of evaluating an expression to a Data_Item. But, suppose a complex criterion resides within a patient state step. Then we would need to have patient state details in the form of a subguideline in order to allow the use of an assignment.

Do we need the assignment action in addition to the Let statement?
Yes, for assigning a new guideline-generated value (e.g., risk) to a data item. Also, we need the assignment action for passing results from decision detail back to the decision step.