

## Tallis Training

---

# Decisions, Candidates and Arguments

---

Ayelet Oettinger

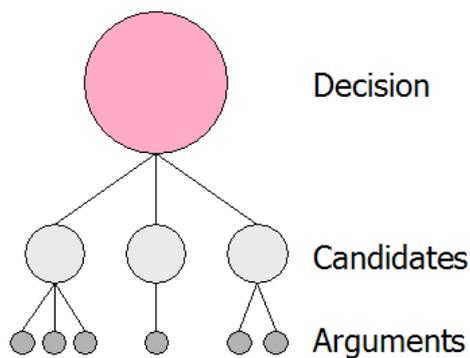
18 July 2005

<b>1. Overview</b>	<b>2</b>
<b>2. Decisions</b>	<b>3</b>
2.1 Inserting a Decision	3
2.2 Decision Attributes	3
<b>3. Candidates</b>	<b>4</b>
3.1 Creating Candidates	4
<b>4. Arguments</b>	<b>4</b>
4.1 Creating Arguments	5
<b>5. Recommendation Rules</b>	<b>6</b>
5.1 Defining a Recommendation Rule	6
<b>6. Handy Examples</b>	<b>6</b>
6.1 Frequently Used Recommendation Rules	6
6.2 Diagnosis Decision	7
6.3 Referral Decision	9

## 1. Overview

Decisions are tasks in which a choice is made – either by the Tallis Engine or by the end-user – between several different options, known as candidates. *PROforma* supports decision-making through a mechanism for generating arguments that may be either *for* or *against* a given candidate.

The figure below describes the structure of a decision: candidates are a property of a decision, and arguments are assigned to each candidate.



An argument is defined by two components:

**Condition** A truth-valued expression that represents the circumstances under which the argument applies (e.g.: `blood_pressure < 120`).

**Support** The support that the argument offers the candidate if the condition is true.

The support format can be either symbolic or numeric:

Symbolic: the argument can be either *for* or *against* the candidate, or it can *confirm* or *exclude* the candidate.

Numeric: the support is defined using a real value; the argument has a weight assigned to it.

When a decision is enacted, the condition and the support properties are used by the Tallis Engine to determine the effect of each argument and in turn the aggregated effect for a given candidate. The aggregated support of all of a candidate's arguments is known as *netsupport*.

A recommendation rule is then used to determine whether or not a particular candidate is recommended. The rule defines an expression that must be true for the candidate to be recommended; this expression typically takes into account the candidate's *netsupport*.

**Note:**

- End-users can select non-recommended candidates if they wish.
- One or more candidates can be selected if the candidate selection mode of the decision is multiple selection.

## 2. Decisions

### 2.1 Inserting a Decision

On the Tallis toolbar are buttons for the different task types:



To insert a decision, click on the purple circle shaped button and then click in the network area to place the decision.

### 2.2 Decision Attributes

By default, decisions are non-automatic. This means that the candidates are displayed to the end-user, who must then select one of them. The candidates are displayed along with an indication as to whether they are recommended by the Tallis Engine or not. If a decision is automatic, the candidates are not displayed to the end-user, and the recommended candidate with the highest netsupport or priority is committed automatically by the Tallis Engine.

#### 2.2.1 Candidates Tab

Apart from the common task attributes, decisions also have a Candidate List: a list of the options to choose from when making a decision.

#### 2.2.2 Sources Tab

Another property of decisions that should be noted is sources. Sources are requests for data. They are typically defined within enquiries, but they can also be defined in the Sources tab of a decision. When defined within a decision, these requests for data have to be fulfilled for the decision to complete. For more information about sources see [Enquiries, Sources, and Data Definitions \(http://www.openclinical.org/TallisTraining/Enquiries\\_Sources\\_and\\_Data\\_Definitions03.htm\)](http://www.openclinical.org/TallisTraining/Enquiries_Sources_and_Data_Definitions03.htm).

#### 2.2.3 Properties Tab

- Candidate Selection: By default, only one candidate can be committed for a given decision. To change this behaviour and allow for multiple selection, select Single Selection in the Candidate Selection field.
- Support Type: Support type is determined per decision. By default, symbolic support is used when defining arguments for candidates. To use numeric support, select Numeric in the Support Type field.

## 3. Candidates

Candidates define the different options for a given decision.

### 3.1 Creating Candidates

To create a new candidate, enter the following information in the Candidates tab:

**Name** An alphanumeric string representing the candidate's ID (e.g.: ReferToGeneticist).

**Description** A text string displayed to the end-user as one of the decision options when the decision is enacted (e.g.: The patient is eligible for referral to a geneticist).

Note: The Dynamic checkbox should be checked if the description contains data items to be evaluated and displayed at runtime.

**Priority** An integer value representing the a-priori importance of the candidate (if several candidates have the same netsupport, candidate priority is used to determine the order in which they are displayed to the end-user).

Note: Although priority is primarily used to order candidates for display, it *can* sometimes affect which candidate gets committed. For example, in an automatic, single selection decision, if more than one candidate is recommended, the one with the highest priority is committed.

When all the data is entered, click Add.

The new candidate will be added to the Candidate List.

Next, arguments and recommendation rules have to be defined for each candidate.

## 4. Arguments

PROforma supports decision-making through a mechanism for generating arguments *for* or *against* a given candidate. When a decision is enacted, the condition and the support properties of the argument are used to determine the effect of each argument and in turn the net support for a given candidate. A recommendation rule is then used to determine whether or not a particular candidate is recommended.

## 4.1 Creating Arguments

To add arguments for (or against) a candidate, select the Arguments tab.

The top drop-down box contains a list of the decision's candidates. Select the candidate for which you want to define arguments, and enter the following information:

**Description** A text string displayed to the end-user during enactment (to represent the argument, e.g.: Blood pressure is lower than 120). If there is no description, the condition is displayed instead.

Note: The Dynamic checkbox should be checked if the description contains data items to be evaluated and displayed at runtime.

**Condition** A truth-valued expression that represents the circumstances under which the argument applies (e.g.: `blood_pressure < 120`)

**Support** The support that the argument offers if the condition is true. The four categories of symbolic support are:

For (+): If the condition is met, the argument offers some support for the candidate.

Against (-): If the condition is met, the argument offers some support against the candidate.

Conf. (++): If the condition is met, the argument conclusively confirms the candidate.

Excl. (--): If the condition is met, the argument conclusively excludes the candidate.

In the Properties tab of the decision properties, support type can be modified from symbolic to numeric. Numeric support allows you to be more precise about the strength of support that an argument offers a candidate by giving each argument a real-valued weight.

When all the data is entered, click Add.

The new argument will be added to the Arguments list.

## 5. Recommendation Rules

When a decision is enacted, the Tallis Engine determines the effect of each argument and in turn the net support for a given candidate. The recommendation rule is then used to determine whether or not a particular candidate is recommended.

### 5.1 Defining a Recommendation Rule

To define a recommendation rule for a candidate, select the Decide tab.

Select the candidate for which you want to define a recommendation rule from the drop-down box.

The Rule field defines an expression that must be true for the candidate to be recommended. It is pre-populated with a default expression:

```
netsupport(CurrentDecision, CurrentCandidate) >= 1
```

The default rule states that for the candidate to be recommended, its aggregated support (or "netsupport") has to be greater than or equal to 1.

Recommendation rules are typically based on the netsupport of the candidate; but any truth-valued expression can serve as a recommendation rule. Recommendation rules can also be based on the netsupport of a competing candidate. For example, in a *Referral* decision which has two candidates, *Refer* and *Do not refer*, the recommendation rule of the *Do not refer* candidate might be:

```
netsupport(ReferralDecision, ReferCandidate) < 1
```

This rule means that the *Do not refer* candidate will only be recommended if the *Refer* candidate's netsupport is less than 1. Note that this rule ignores the netsupport of the *Do not refer* candidate itself.

**Note:** Selecting Unlocked in the Rule Input Mode field will reset the recommendation to the default rule.

## 6. Handy Examples

### 6.1 Frequently Used Recommendation Rules

For the following examples, assume a decision named *decision\_1* with three candidates: *candidate1*, *candidate2* and *candidate3*.

- Recommendation rule for recommending candidate1 if it is supported:

```
netsupport(decision_1, candidate1) >= 1
```

- Recommendation rule for recommending candidate1 if none of the other candidates are supported:

```

netsupport(decision_1, candidate2) < 1 AND
netsupport(decision_1, candidate3) < 1

```

- Recommendation rule for recommending candidate1 if it has the highest net support:

```

netsupport(decision_1, candidate1) >
netsupport(decision_1, candidate2) AND
netsupport(decision_1, candidate1) >
netsupport(decision_1, candidate3)

```

## 6.2 Diagnosis Decision

In the Cold and Flu Guide sample, a *Diagnosis* decision is made between the options of *cold* and *flu*. The decision is based on the table below.

Symptoms	Cold	Flu
<i>Fever</i>	Rare	Characteristic, high (102-104F)
<i>Headache</i>	Rare	Prominent
<i>General aches, pains</i>	Slight	Usual; often severe
<i>Extreme exhaustion</i>	Never	Early and prominent
<i>Stuffy nose</i>	Common	Sometimes
<i>Sore throat</i>	Common	Sometimes
<i>Cough</i>	Mild to moderate	Common; can become severe

\*Nasal spray flu vaccine contains weakened flu viruses and can occasionally induce flu

The *Diagnosis* decision has three candidates:

- Flu
- Cold
- Other

## 6.2.1 Flu

The arguments for flu can be divided into 3 groups:

- a. Arguments that provide some support (i.e., arguments *for*):
  - Severe cough
  - Recent nasal spray flu vaccine
  - Extreme exhaustion
  - General aches and pains
  - Headache
- b. Confirming arguments:
  - Fever
- c. Excluding arguments:
  - No fever

```
Recommendation Rule: netsupport(DiagnosisDecision, Flu) >=
netsupport(DiagnosisDecision, Cold) AND
netsupport(DiagnosisDecision, Flu) >= 1
```

The recommendation rule for flu states that flu would only be recommended if two conditions are met:

- a. Net support for flu is higher than or equal to 1 – meaning there's at the very least some argument for flu.
- b. Net support for flu is higher than or equal to the net support for cold – meaning there's some argument for both, but more support for flu.

## 6.2.2 Cold

The arguments for cold can be divided into 2 groups:

- a. Arguments that provide some support (i.e., arguments *for*):
  - Mild to moderate cough
  - Runny stuffy nose
  - Sore throat
- b. Arguments that are somewhat against (i.e., arguments *against*):
  - Extreme exhaustion
  - Headache

- Fever

Recommendation Rule: `netsupport(DiagnosisDecision, Cold) >= netsupport(DiagnosisDecision, Flu) AND netsupport(DiagnosisDecision, Cold) >= 1`

The recommendation rule for cold states that cold would only be recommended if two conditions are met:

- a. Net support for cold is higher or equal to 1 – meaning there's at the very least some argument for cold.
- b. Net support for cold is higher or equal to the net support for flu – meaning there's some argument for both, but more support for cold.

### 6.2.3 Other

The Other candidate has no arguments.

Recommendation Rule: `netsupport(DiagnosisDecision, Flu) < 1 AND netsupport(DiagnosisDecision, Cold) < 1`

This rule states that the *Other* candidate is only recommended if the net support for the other two candidates is less than 1. So it is only recommended if there's no support for either cold or flu.

## 6.3 Referral Decision

The decision has three candidates:

- Two week referral
- Non urgent referral
- No referral

### 6.3.1 Two week referral

Arguments for *Two week referral*:

- Nipple retraction or distortion
- Eczema
- Skin distortion
- Skin nodule
- Skin ulceration
- Discrete lump in a woman over 30

Recommendation rule: `Netsupport( Referral_decision_breast, Two_week_referral ) >= 1`

According to this rule, if any of the above arguments are true, a two-week referral is recommended.

### 6.3.2 Non urgent referral

Arguments for *Non urgent referral*:

- Asymmetrical nodularity
- Intractable pain
- Nipple discharge in a patient over 50
- Large volume, bilateral nipple discharge
- Bloodstained nipple discharge
- Abscess
- Cyst
- Discrete lump in a patient under 30

Recommendation rule: `Netsupport( Referral_decision_breast, Non_urgent_referral ) >= 1 AND netsupport( Referral_decision_breast, Two_week_referral ) < 1`

According to this rule, if any of the above arguments are true *and* if there is no support for a two-week referral, a non-urgent referral is recommended.

### 6.3.3 No referral

The *No referral* candidate has no arguments.

Recommendation rule: `Netsupport( Referral_decision_breast, Two_week_referral ) < 1 and Netsupport( Referral_decision_breast, Non_urgent_referral ) <1`

This rule states that if a two-week referral and a non-urgent referral are not supported, no referral is recommended.