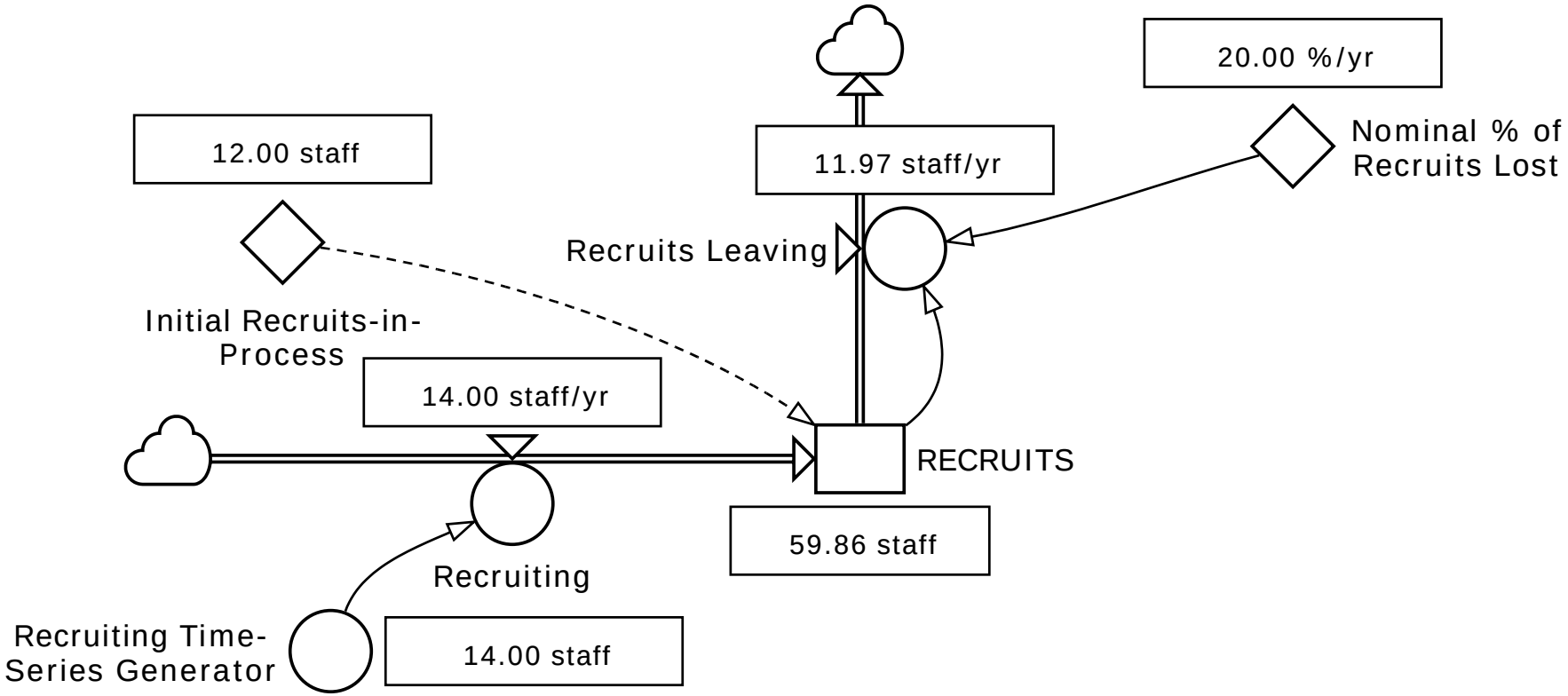
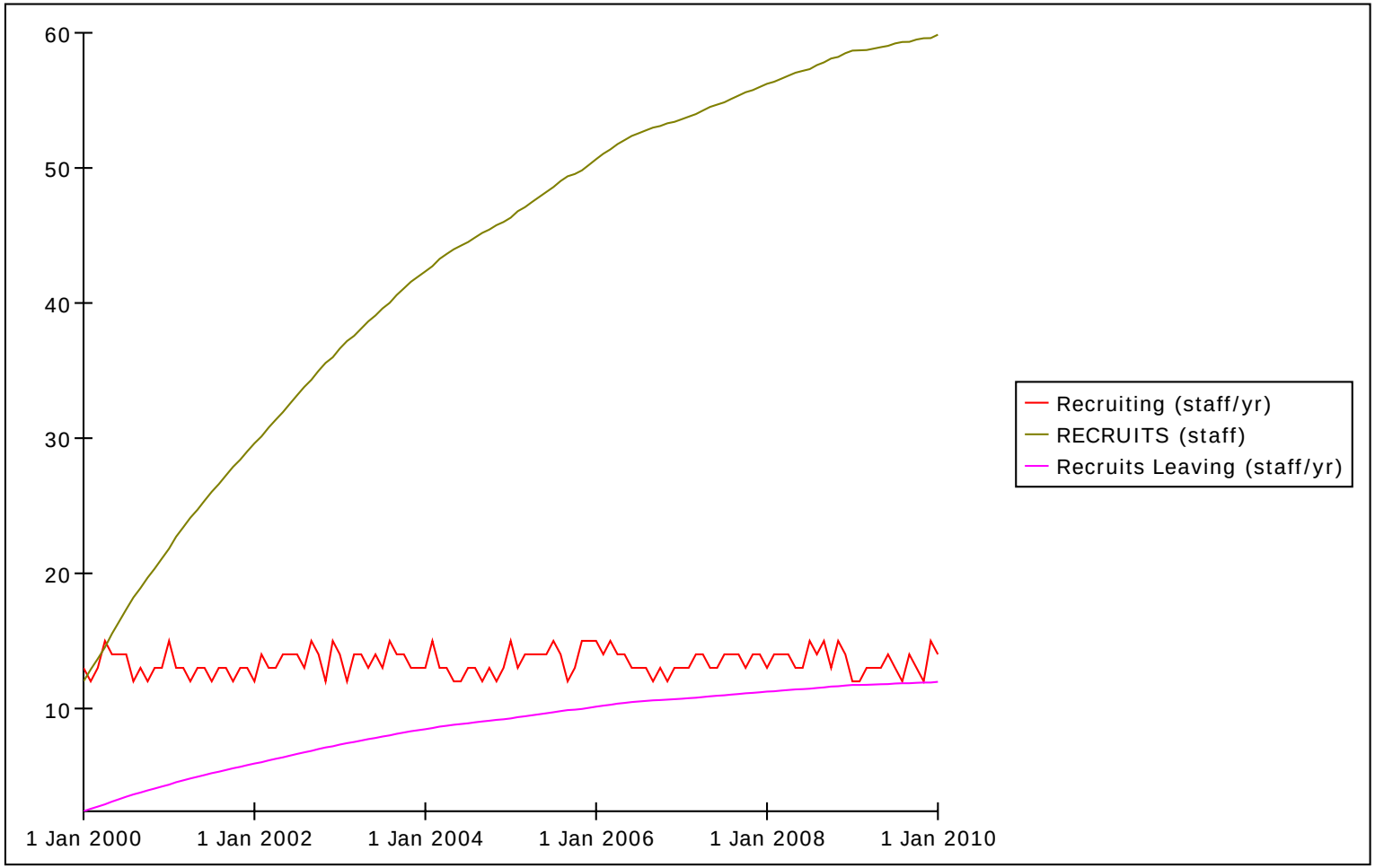
FUNCTIONAL DESCRIPTION
People identified as being suitable enter a process which takes an unspecified time, during which they complete their training and formalities associated with becoming a Simultech employee. Whilst in the training and recruiting process they are designated as RECRUITS. They remain so designated until they leave this training and recruiting process. A nominal percentage of RECRUITS will leave every year. The leaving individuals will not complete their training nor the necessary recruiting formalities. For example, they might choose to take up alternate employment. Those who do not leave through attrition will accumulate as RECRUITS.

PHYSICAL DESCRIPTION
The simulation timestep (dt) is set at 1 month, which is one sixth of the shortest delay expected in the model. The simulation starts in January 2001 and stops January 2010 even though the period of interest is July 2005 to July 2007. A representative set of time-series data is generated as input to the recruiting process. The auxiliary variable 'Recruiting Time-Series Generator' generates typical numbers of recruits for each timestep of the simulation. These numbers vary randomly between 12.00 and 15.00 and are rounded to produce integers, having values 12, 13, 14, or 15. This variable is defined as: ROUND(RANDOM(12,15,0.9)*1<<staff>>, 1<<staff>>, 1<<staff>>). A nominal 20% per year are lost from the stock of rookies accumulating. The initial number of rookies is 12 (a number chosen arbitrarily) and this number is added to each month by the 'Recruiting' rate variable. Attrition at the rate of 20% per year limits the continued growth of the stock of rookies. The number leaving in any year is: (RECRUITS*'Nominal % of Recruits Lost'<<staff/yr>>>)

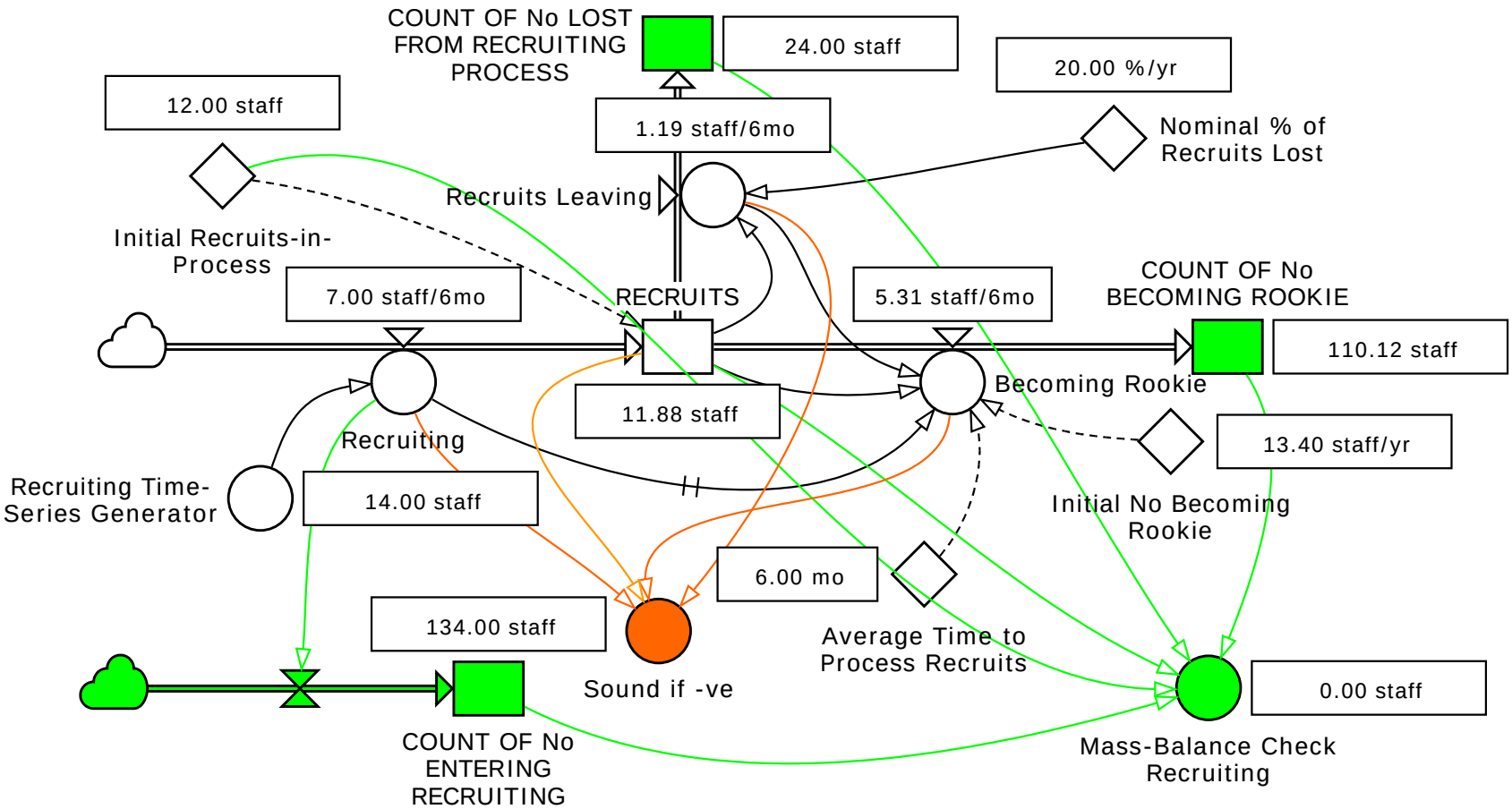
INTERFACES
The mature version of this functional module will take input from a combination of an exogenous force in the form of demand for senior consultants and feedback produced by the extant gap between target numbers of staff and actual numbers. However, in this version; input from the outside world as controlled by this exogenous force and the feedback described is represented by random numbers. Loss to the outside world only occurs through attrition. In subsequent modelling iterations RECRUITS will progress to become ROOKIES.

VERIFICATION
Before proceeding, the model must be verified using a range of tests described in McLucas, A.C., 2005, 'System Dynamics Application: A Modular Approach to Modelling Complex World Behaviour', Argos Press, Canberra: Ch 6.


20.00 %/yr
12.00 staff
11.97 staff/yr
Nominal % of Recruits Lost
Recruits Leaving
Initial Recruits-in-Process
14.00 staff/yr
RECRUITS
59.86 staff
Recruiting
Recruiting Time-Series Generator
14.00 staff



Because there is no outflow from the stock of RECRUITS, other than attrition, the stock continues to grow until numbers being lost (20% of the total number of recruits in any one year) are sufficient to match the inflow (in the range of 12 to 15 per year). To see this effect more clearly, it would be necessary to extent the time horizon by several years, say, to 2020.



The symbols added to enable the mass-balance test to be conducted are shown in green. The value of Mass-Balance Check Recruiting is found to return 0.00 at the beginning and end of the simulation, and for every timestep of the simulation.

The symbols added to provide an alarm if stock of RECRUITS or any of the flows become negative are shown in orange.

This model combines the functionality of the models developed in previous steps. It is stressed that the numbers entering the recruiting process in subsequent versions of the model might be different to those artefically created here. The primary purpose of each interim model is to demonstrate functionality, which must be verified by appropriate tests. Strictly speaking, all interim models produced will be 'invalid' insofar as they do not acually represent the numbers we will find in the real-world problem. Further, they are only partial representations of real-world behaviour. It is only when we create progressively more mature models that we represent the real-world behaviour. Further, testing to validate models cannot be attempted until models are relatively mature.

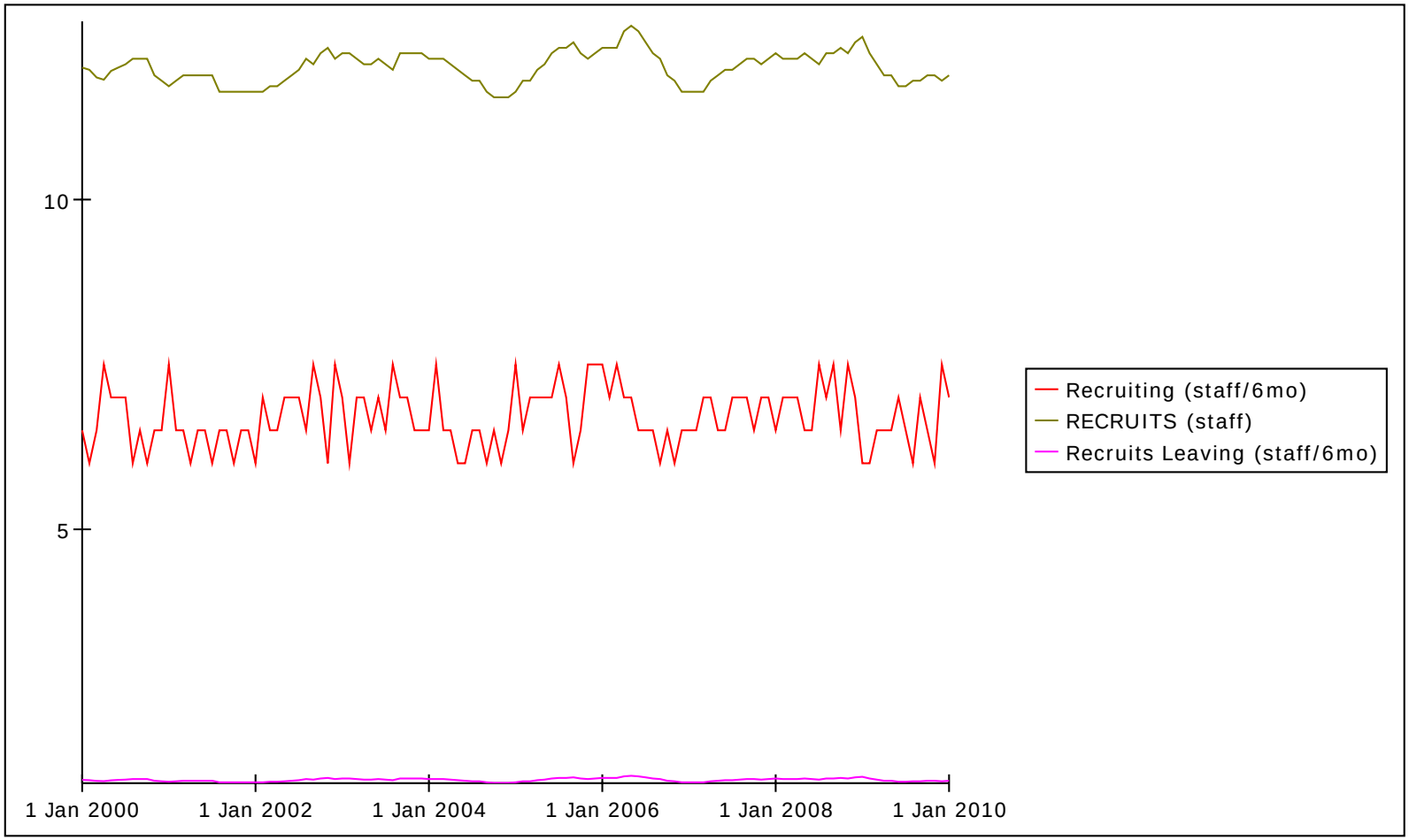
Those personnel entering the recruiting process remain for a time equal to the 'Average Time to Process Recruits', in a pipeline delay. That is, using a conveyor belt methaphor, those placed on the (recruiting process) conveyor belt remain there for a set time (detemined by the length of the conveyor belt). However, individuals might leave. This is the same as plucking individuals off the conveyor belt, with the consequence that some do not reach the end. The logical structure of the model must be checked to assure that double-counting does not occur: those plucked off (leave the recruiting process before that process is complete) cannot be promoted to enter the subsequent stock of ROOKIES (which we are yet to create).

It is not necessary to create an integer model, though to do so might appear more rational and intuitive. This model produces representative numbers which (it might readily be argued) do not make complete sense. This is true in the short run, or on the basis of any single step in the simulation. However, when simulation runs become relatively long, or numbers of individuals being modelled becomes large, then the apparent need for integer modelling becomes far less important (if it was ever a real requirement).

In any ageing chain (pipeline) model, it is important to ensure that undesirable transient effects are not created because the pipelines are initially empty. To avoid this, representative initial numbers are provided as input to the appropriate rate-controlling variables.

Functionality should always checked in the first instance by analysis of tabulated results for each timestep. Note that the simulation Start Time is set to a time much earlier than the current date. This provides an opportunity for tranient effects to play out fully. We might have chosen to set the Start Time back by a period equivalent to the longest delay in this model. However, as this is being built as a transitional model we need to make an estimate of the longest delay in the mature version of the model. That delay, for experienced staff to become experts is five years, so we commence our simulation at the beginning of 2000 (assuming today's date is 1 July 2005). Similarly, we need to project our simulation time-horizon beyond the period of immediate interest. We do this to enable us to identify any possible long-term instability in our model.

Time	RECRUITS	Recruiting	Recruits Leaving	Becoming Rookie
1 Jan 2000	12.00 staff	6.50 staff/6mo	1.20 staff/6mo	5.50 staff/6mo
1 Feb 2000	11.97 staff	6.00 staff/6mo	1.20 staff/6mo	5.50 staff/6mo
1 Mar 2000	11.85 staff	6.50 staff/6mo	1.19 staff/6mo	5.52 staff/6mo
1 Apr 2000	11.82 staff	7.50 staff/6mo	1.18 staff/6mo	5.52 staff/6mo
1 May 2000	11.95 staff	7.00 staff/6mo	1.20 staff/6mo	5.51 staff/6mo
1 Jun 2000	12.00 staff	7.00 staff/6mo	1.20 staff/6mo	5.50 staff/6mo
1 Jul 2000	12.05 staff	7.00 staff/6mo	1.21 staff/6mo	5.30 staff/6mo
1 Aug 2000	12.13 staff	6.00 staff/6mo	1.21 staff/6mo	4.79 staff/6mo
1 Sep 2000	12.13 staff	6.50 staff/6mo	1.21 staff/6mo	5.29 staff/6mo
1 Oct 2000	12.13 staff	6.00 staff/6mo	1.21 staff/6mo	6.29 staff/6mo
1 Nov 2000	11.88 staff	6.50 staff/6mo	1.19 staff/6mo	5.81 staff/6mo
1 Dec 2000	11.80 staff	6.50 staff/6mo	1.18 staff/6mo	5.82 staff/6mo
1 Jan 2001	11.72 staff	7.50 staff/6mo	1.17 staff/6mo	5.83 staff/6mo
1 Feb 2001	11.80 staff	6.50 staff/6mo	1.18 staff/6mo	4.82 staff/6mo
1 Mar 2001	11.88 staff	6.50 staff/6mo	1.19 staff/6mo	5.31 staff/6mo
1 Apr 2001	11.88 staff	6.00 staff/6mo	1.19 staff/6mo	4.81 staff/6mo
1 May 2001	11.88 staff	6.50 staff/6mo	1.19 staff/6mo	5.31 staff/6mo
1 Jun 2001	11.88 staff	6.50 staff/6mo	1.19 staff/6mo	5.31 staff/6mo
1 Jul 2001	11.88 staff	6.00 staff/6mo	1.19 staff/6mo	6.31 staff/6mo
1 Aug 2001	11.63 staff	6.50 staff/6mo	1.16 staff/6mo	5.34 staff/6mo
1 Sep 2001	11.63 staff	6.50 staff/6mo	1.16 staff/6mo	5.34 staff/6mo
1 Oct 2001	11.63 staff	6.00 staff/6mo	1.16 staff/6mo	4.84 staff/6mo
1 Nov 2001	11.63 staff	6.50 staff/6mo	1.16 staff/6mo	5.34 staff/6mo
1 Dec 2001	11.63 staff	6.50 staff/6mo	1.16 staff/6mo	5.34 staff/6mo

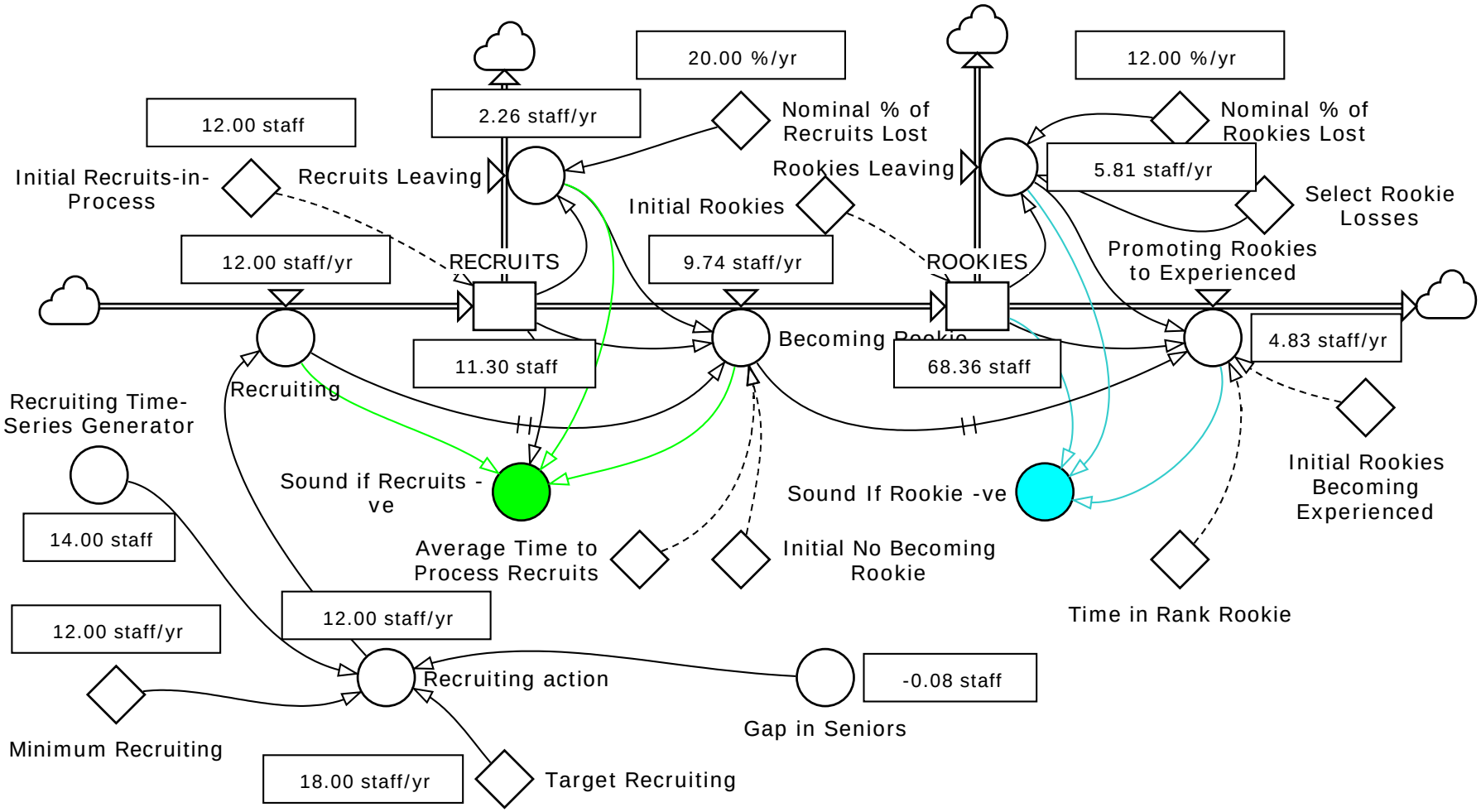


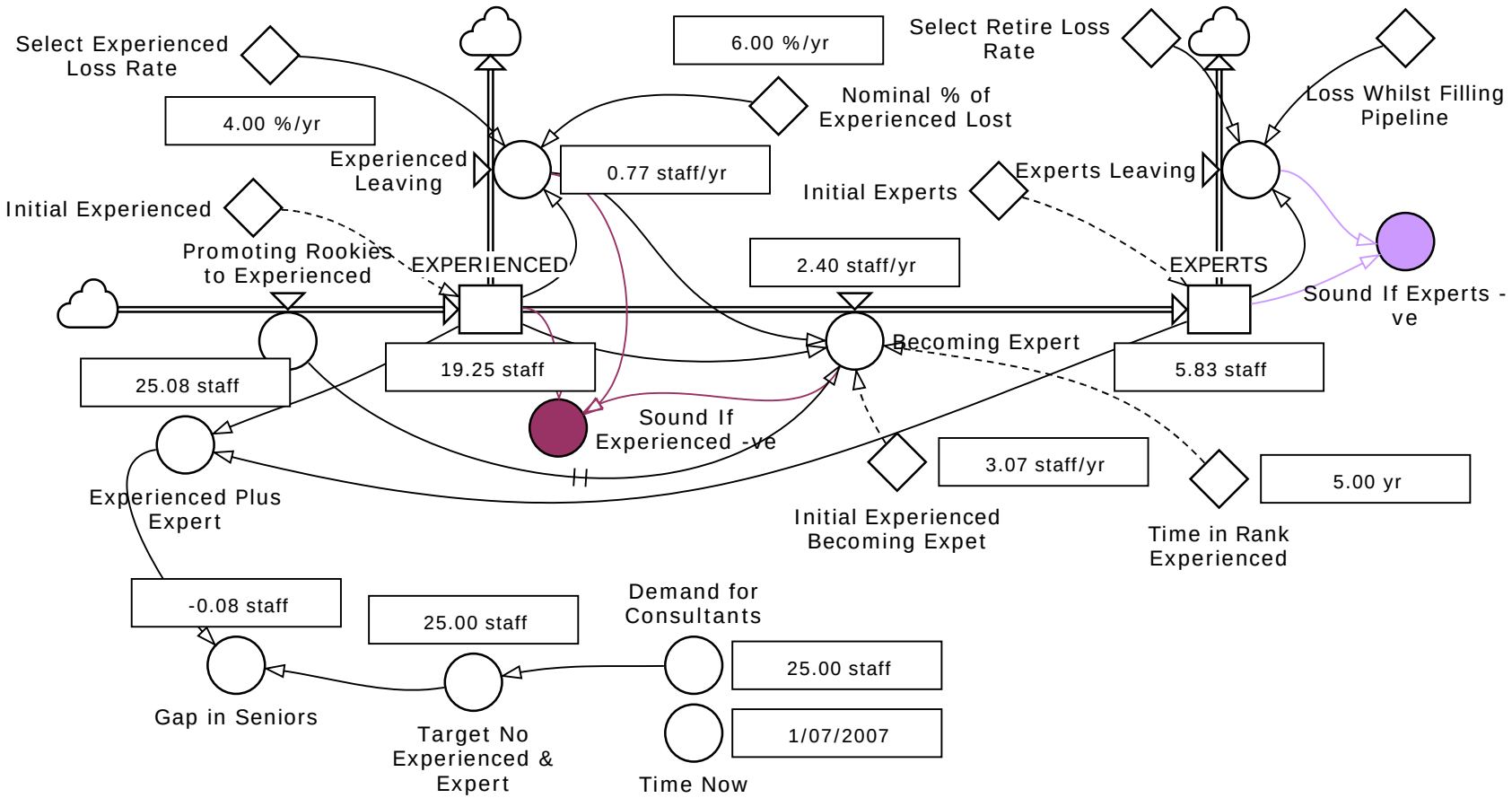
Note that for the initial period equivalent to the 'Average Time to Process Recruits', the pipeline is being arteficially filled at the rate of the chosen initial value 'Initial No Becoming Rookie'. This is a convenient way of representing flow that does not result from the input rate-controlling variable. However, it is necessary to avoid the otherwise unintended build up of the stock RECRUITS. To demonstrate this, set 'Initial No Becoming Rookie' to zero and re-run the simulation.

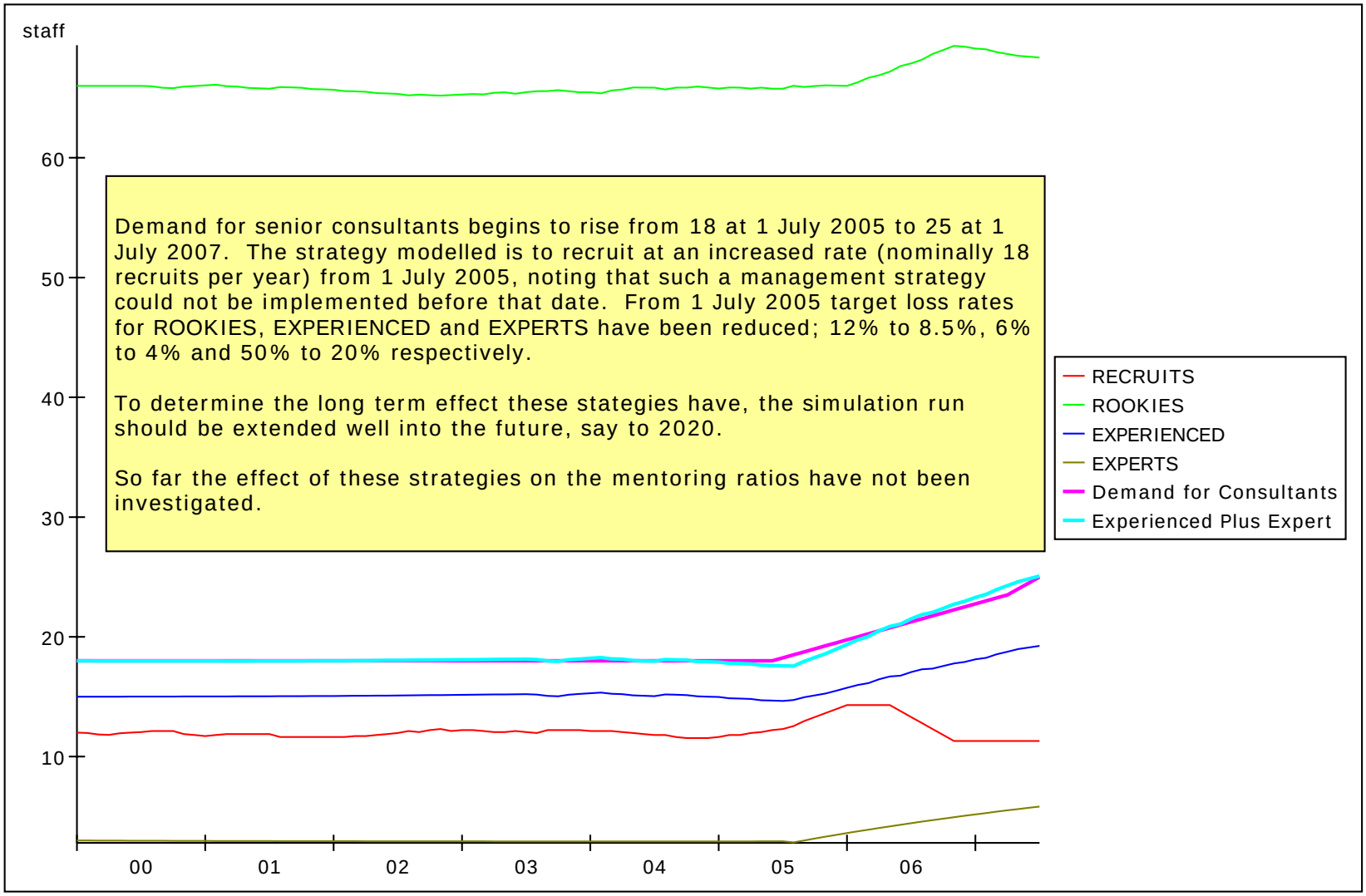
```
mainmodel Component 1 {
    const Average Time to Process Recruits {
        autotype Real
        unit mo
        init 6
    }
    aux Becoming Rookie {
        autotype Real
        unit staff/6mo
        def IF(RECRUITS>0<<staff>>,1,0)
            *DELAYPPL(Recruiting,
            'Average Time to Process Recruits',
            'Initial No Becoming Rookie')
            -'Recruits Leaving'
        doc When formulating the definition for this rate-controlling variable we must appreciate that for the initial delay period, the value of the outflow is determined by 'Initial
            No Becoming Rookie'.  This number is chosen to ensure an outflow of similar magnitude to that which occurs after the initial delay period has passed.  Regardless
            of the value of 'Initial No Becoming Rookie' chosen, Powersim Studio will subtract from it the value of 'Recruits Leaving' for the current step in the simulation.
            Hence, if 'Initial No Becoming Rookie' is less than 'Recruits Leaving' then 'Becoming Rookie' will take on a negative value.  Clearly this is not realistic, though it is a
            consequence of the formulation of the definition of 'Becoming Rookie'.  Rather than attempting to create guard statements to preclude this, an alarm is set up to
            warn us if 'Becoming Rookie' takes on negative values.  If this occurs, then the value of 'Initial No Becoming Rookie' needs to be adjusted accordingly.
        note IF(RECRUITS>0<<staff>>,1,0)
            *DELAYPPL(Recruiting,
            'Average Time to Process Recruits',
            'Initial No Becoming Rookie')
            -'Recruits Leaving'
    }
    level COUNT OF No BECOMING ROOKIE {
        autotype Real
        unit staff
        init 0
        inflow { autodef  'Becoming Rookie' }
    }
    level COUNT OF No ENTERING RECRUITING {
        autotype Real
        unit staff
        init 0
        inflow { autodef  Recruiting }
    }
    level COUNT OF No LOST FROM RECRUITING PROCESS {
        autotype Real
        unit staff
        init 0
        inflow { autodef  'Recruits Leaving' }
```

```
}
const Initial No Becoming Rookie {
    autotype Real
    unit staff/yr
    init 13.4<<staff/yr>>
}
const Initial Recruits-in-Process {
    autotype Real
    unit staff
    init 12
}
aux Mass-Balance Check  Recruiting {
    autotype Real
    autounit staff
    def 'Initial Recruits-in-Process'
        -RECRUITS+
        'COUNT OF No ENTERING RECRUITING'
        -'COUNT OF No LOST FROM RECRUITING PROCESS'
        -'COUNT OF No BECOMING ROOKIE'
    doc The mass-balance test described at McLucas, A.C., 2005, 'System Dynamics Application: A Modular Approach to Modelling Complex World Behaviour', Argos
        Press, Canberra: 164-165, is applied to verify that the model does not contain errors which lead to either an accumulation or loss.
}
const Nominal % of Recruits Lost {
    autotype Real
    unit %/yr
    init 20
}
aux Recruiting {
    autotype Real
    unit staff/6mo
    def 'Recruiting Time-Series Generator'/1<<yr>>
    doc If it is possible that the rate-controlling variable might erroneously take on negative values (in the case that 'Recruiting Time-Series Generator, or an auxiliary which
        will subsequently replace it when the model is further developed), it might be necessary to multiply the definition of the 'Recruiting' variable by
        IF(Recruiting Time-Series Generator>0,1,0).  See Note below for a re-defined 'Recruiting' rate-controlling auxiliary variable.
        To provide an alarm if input to recruiting numbers becomes negative, two separate auxiliary variables might be created to monitor the status of the 'Recruiting'
        variable.  See Note below for an example of how this might be formulated.
    note 'Recruiting Time-Series Generator'/1<<mo>>*IF('Recruiting Time-Series Generator'>0<<staff>>,1,0)
        STOPIF(Recruiting<0<<staff/mo>>)
        SOUND(IF(Recruiting<0<<staff/mo>>,1,0,0))
}
aux Recruiting Time-Series Generator {
    autotype Real
    unit staff
```

```
        def ROUND(RANDOM(12,15,0.9)*1<<staff>>, 1<<staff>>, 1<<staff>>)
        doc This auxiliary variable generates real numbers having random values between 12.00 and  15.00.  These numbers are then rounded to the nearest integer value.
            The values returned are 12,13,14, or 15 staff, which will be provided as input to the rate-controlling variable 'Recruiting'.  This auxiliary variable will produce these
            numbers for as long as the simulation is run.
            An alternate formulation might take the form of a graphical input for a very specific set number of periods, in this case the first 96 one-month periods:
            GRAPH((TIME-STARTTIME),1<<mo>>,1<<mo>>,{
            12,14,12,15,12,13,14,13,15,15,15,15,12,14,14,14,15,12,15,12,12,12,14,15,
            13,13,14,15,13,15,15,15,15,12,14,14,14,15,14,12,13,15,13,13,13,13,13,14,
            15,12,14,14,14,15,12,15,12,12,12,14,15,15,12,14,14,14,15,12,15,12,12,12,
            14,15,13,13,14,13,14,12,15,15,12,13,14,15,12,14,14,14,15,12,15,12,12,12,
            }*1<<pers>>)
        note The main purpose here is to develop a model with the required functionality.  The model can then be progressively 'tuned' to replicate real-world behaviour.
    }
    level RECRUITS {
        autotype Real
        unit staff
        init 'Initial Recruits-in-Process'
        inflow { autodef  Recruiting }
        outflow { autodef  'Recruits Leaving' }
        outflow { autodef  'Becoming Rookie' }
    }
    aux Recruits Leaving {
        autotype Real
        unit staff/6mo
        def RECRUITS*'Nominal % of Recruits Lost'
    }
    aux Sound if -ve {
        autotype Real
        def SOUND(IF(Recruiting<0<<staff/mo>>,1,0,0))+
            SOUND(IF('Becoming Rookie'<0<<staff/mo>>,1,0,0))+
            SOUND(IF(RECRUITS<0<<staff>>,1,0,0))+
            SOUND(IF('Recruits Leaving'<0<<staff/mo>>,1,0,0))
        doc This auxiliary variable has been created to sound an alarm if the stock of rookies or flows in the model become negative.
    }
}
unit staff {
    def ATOMIC
    doc This is a generic unit for individual people.  No discrimination is made between individuals.
}
```


20.00 %/yr
12.00 %/yr
12.00 staff
2.26 staff/yr
Nominal % of Recruits Lost
Nominal % of Rookies Lost
Initial Recruits-in-Process
Recruits Leaving
Rookies Leaving
5.81 staff/yr
Select Rookie Losses
Initial Rookies
12.00 staff/yr
RECRUITS
9.74 staff/yr
ROOKIES
Promoting Rookies to Experienced
Becoming Rookie
11.30 staff
68.36 staff
4.83 staff/yr
Recruiting
Recruiting Time-Series Generator
Sound if Recruits -ve
Sound If Rookie -ve
Initial Rookies Becoming Experienced
14.00 staff
Average Time to Process Recruits
Initial No Becoming Rookie
Time in Rank Rookie
12.00 staff/yr
12.00 staff/yr
Recruiting action
-0.08 staff
Minimum Recruiting
Gap in Seniors
18.00 staff/yr
Target Recruiting


Select Experienced Loss Rate
4.00 %/yr
6.00 %/yr
Nominal % of Experienced Lost
Experienced Leaving
0.77 staff/yr
Select Retire Loss Rate
Loss Whilst Filling Pipeline
Initial Experienced
Promoting Rookies to Experienced
EXPERIENCED
2.40 staff/yr
Initial Experts
Experts Leaving
EXPERTS
Sound If Experts - ve
Becoming Expert
25.08 staff
19.25 staff
5.83 staff
Sound If Experienced -ve
Experienced Plus Expert
3.07 staff/yr
Initial Experienced Becoming Expet
5.00 yr
Time in Rank Experienced
-0.08 staff
25.00 staff
Demand for Consultants
25.00 staff
Gap in Seniors
Target No Experienced & Expert
1/07/2007
Time Now


staff
60
50
40
30
20
10
00
01
02
03
04
05
06
Demand for senior consultants begins to rise from 18 at 1 July 2005 to 25 at 1 July 2007. The strategy modelled is to recruit at an increased rate (nominally 18 recruits per year) from 1 July 2005, noting that such a management strategy could not be implemented before that date. From 1 July 2005 target loss rates for ROOKIES, EXPERIENCED and EXPERTS have been reduced; 12% to 8.5%, 6% to 4% and 50% to 20% respectively.
To determine the long term effect these stategies have, the simulation run should be extended well into the future, say to 2020.
So far the effect of these strategies on the mentoring ratios have not been investigated.
RECRUITS
ROOKIES
EXPERIENCED
EXPERTS
Demand for Consultants
Experienced Plus Expert
Current Time
1/07/2007
25.08 staff
Experienced Plus Expert