Woodstock Technical Bulletin

Steven Northway, Corporate Forestry, MacMillan Bloedel Ltd.

Steve Northway submitted this innovative modeling solution. If you had asked us before if variable time steps were possible in Woodstock, we would have said probably not – we were wrong.

MODELING VARIABLE TIME STEPS IN WOODSTOCK

Continuous time systems are modeled with discrete time steps for computational ease, but with the intent of capturing the behavior of the original system. A small time step might capture the behavior of the system more accurately, but at the cost of higher computational demands. An elegant answer is to make use of variable time steps. Here, small time steps are used only where they are necessary to capture details of system behavior.

Large time steps are adequate for capturing the behavior of a system that has reached a steady state, but during times of rapid change or dynamic interactions of system components short time steps are necessary to capture system behavior. There is also an advantage in longer time steps in conjunction with a less precise description of the system in reducing computational complexity.

In landscape modeling, the early part of the planning horizon is usually more dynamical and the information needs more heightened then in the later part. As an example, a one-year delay in operability is more likely to be a real constraint in the first year of the plan then in the last. It might also be unacceptable to model adjacency constraints in the early part of the plan to other than the year, while the nearest five-year period would do towards the end of the plan. As a consequence it makes sense to begin with short periods and progress to longer ones.

The flexibility of Woodstock makes it possible to include different time steps in the same model. The key is to provide Woodstock with a set of near duplicate themes, differing only in the length of period they represent (e.g., hw_1 and hw_10 representing hardwoods in one and ten-year periods respectively). Each theme needs to be provided with an appropriate set of yields, life spans, transitions, outputs, constraints, etc., consistent with their period length. Then, at some predefined period Woodstock needs to be instructed to re-summarize (TRANSITION) the existing period-length themes into ones corresponding to the new period-length (e.g., from hw_1 to hw_10).

The easiest way to convey all this is to look at a simple, but not trivial, example. It starts off with 20 1-year periods followed by 20 10-year periods.

This example demonstrates an approach for modeling variable time steps within Woodstock.

File	Comments	Syntax
1.PRI	Nothing special is required in the primary file.	CONTROL *LENGTH 40 *QUEUE OFF *OPTIMIZE OFF *SCHEDULE ON *PAUSE 5 LANDSCAPE [1.LAN] AREAS [1.ARE] LIFESPAN [1.LIF] YIELDS [1.YLD] ACTIONS [1.ACT] TRANSITIONS [1.TRN] OUTPUTS [1.OUT] OPTIMIZE [1.OPT] SCHEDULE [1.SEQ] REPORTS [1.RPT] GRAPHICS [1.GRA]
1.LAN	The landscape file includes a version of each Stand type for representation in 1 and 10-year periods.	*THEME Stand type og existing old growth with static volume nat natural stands 1 yr classes plant planted stands 1 yr classes og_10 old growth stands 10yr classes nat_10 natural stands 10yr class plant_10 planted stands 10 yr classes *AGGREGATE ONE og nat plant *AGGREGATE TEN og_10 nat_10 plant_10
1.ARE	The areas file only includes areas for the Stand types representing 1-year periods.	*A og 80 40 *A plant 03 3 *A plant 06 3 *A plant 09 3 *A plant 12 2 *A plant 15 2 *A plant 18 2 *A plant 21 1 *A plant 24 1

```
*A plant 27 1
*A plant 30 1
*A nat 03 1
*A nat 06 1
*A nat 09 1
*A nat 12 2
*A nat 15 2
*A nat 18 2
*A nat 21 3
*A nat 24 3
*A nat 27 3
*A nat 30 3
*A nat 32 2
*A nat 34 2
*A nat 36 2
*A nat 38 2
*A nat 40 2
*A nat 42 2
*A nat 44 2
*A nat 46 2
*A nat 48 2
*A nat 50 2
```

1.LIF These species are long lived, so it makes use of an arbitrarily large number.

```
? 600
```

1.YLD There is a yield table for each of the stand types. The ones representing 1-year periods have annual steps, and the ones representing 10-year periods represent the average of a 10-year period.

```
*Y og
  m3 1
    700
*Y og_10
  m3 1
    700
*Y nat
  m3 1
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 1 1
    2 4 9 14 19 24 36 51 66 81
    96 114 133 151 170 188 206 225 243 261
    279 296 313 330 347 364 380 396 412 430
    448 465 483 500 516 533 548 563 577 591
    605 619 632 645 660 674 688 702 716 730
    743 757 770 783 797 811 825 838 851 864
    877 890 902 915 925 936 947 957 968 978
    988 998 1008 1017 1027 1037 1047 1057
1067 1077
*Y nat_10 ; yields are averages of 10 years
  m3 1
```

```
    0 0 5 92 269 439 597 736 870 982 1067
1077
*Y plant
  m3 1
      0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 1 2 3 4 9
    17 26 35 43 59 78 96 114 132 150
    169 188 206 224 242 260 278 296 314 332
    349 368 386 404 422 439 456 477 497 517
    537 556 576 594 612 629 646 663 679 695
    711 727 744 762 780 797 814 831 848 865
    881 897 913 929 945 960 975 990 1005
1020
    1034 1048 1060 1072 1084 1095 1107 1118
1129 1140
    1151 1161 1172 1183 1195 1207 1219 1231
1242 1254
*Y plant_10
  m3 1
    0 0 20 160 341 526 704 872 1025 1145
1242 1254

*YC og
  pi _DELTA(m3,-1)
*YC nat
  pi _DELTA(m3,-1)
*YC plant
  pi _DELTA(m3,-1)
*YC plant_10
  pi _DELTA(m3,-1)
*YC nat_10
  pi _DELTA(m3,-1)
*YC og_10
  pi _DELTA(m3,-1)
```

1.ACT

It is in the use of the action _DEATH that the switch to ten-year periods happens. A purpose made action could be used, but the use of _DEATH has the advantage of presenting the LP with a compact problem. As _DEATH is 'known' to Woodstock it knows not to provide for the possibilities of the one-year stand types in the ten-year periods.

_DEATH was redefined using time-dependent

```
*ACTION cc Y clear cut
  *OPERABLE cc
    og _AGE 1 _AGE 600
    nat _AGE 40 _AGE 600
    plant _AGE 40 _AGE 600
    og_10 _AGE 1 _AGE 600
    nat_10 _AGE 4 _AGE 600
    plant_10 _AGE 4 _AGE 600

*ACTION _DEATH Y
  *OPERABLE _DEATH _CP 1..19
    ? _AGE = 600
  *OPERABLE _DEATH _CP 20
    ONE _AGE >= 0
    TEN _AGE  = 600
```

	using time-dependent operabilities. In period 20, all 1-year types must 'die', while 10-year types die at lifespan. In all other periods, types die at lifespan.	`*OPERABLE _DEATH _CP 21.._LENGTH` `  ? _AGE = 600`
1.TRN	It is the transfer of one-year period stands types to ten-year period stand types that accomplishes the re-summarization. This is accomplished through defining _DEATH.	see code below

```
*CASE cc
  *SOURCE og
    *TARGET plant 80
    *TARGET nat   20
  *SOURCE nat
    *TARGET plant 80
    *TARGET nat   20
  *SOURCE plant
    *TARGET plant 100
  *SOURCE og_10
    *TARGET plant_10 80
    *TARGET nat_10   20
  *SOURCE nat_10
    *TARGET plant_10 80
    *TARGET nat_10   20
  *SOURCE plant_10
    *TARGET plant_10 100

*CASE _DEATH
  *SOURCE nat @1..10
    *TARGET nat_10 100 _AGE 1
  *SOURCE nat @11..20
    *TARGET nat_10 100 _AGE 2
  *SOURCE nat @21..30
    *TARGET nat_10 100 _AGE 3
  *SOURCE nat @31..40
    *TARGET nat_10 100 _AGE 4
  *SOURCE nat @41..50
    *TARGET nat_10 100 _AGE 5
  *SOURCE nat @51..60
    *TARGET nat_10 100 _AGE 6
  *SOURCE nat @61..70
    *TARGET nat_10 100 _AGE 7
  *SOURCE nat @71..80
    *TARGET nat_10 100 _AGE 8
  *SOURCE nat @81..90
    *TARGET nat_10 100 _AGE 9
  *SOURCE nat @91..120
    *TARGET nat_10 100 _AGE 10
  *SOURCE nat
    *TARGET nat_10 100 _AGE 1

  *SOURCE plant @1..10
    *TARGET plant_10 100 _AGE 1
```

```
  *SOURCE plant @11..20
    *TARGET plant_10 100 _AGE 2
  *SOURCE plant @21..30
    *TARGET plant_10 100 _AGE 3
  *SOURCE plant @31..40
    *TARGET plant_10 100 _AGE 4
  *SOURCE plant @41..50
    *TARGET plant_10 100 _AGE 5
  *SOURCE plant @51..60
    *TARGET plant_10 100 _AGE 6
  *SOURCE plant @61..70
    *TARGET plant_10 100 _AGE 7
  *SOURCE plant @71..80
    *TARGET plant_10 100 _AGE 8
  *SOURCE plant @81..90
    *TARGET plant_10 100 _AGE 9
  *SOURCE plant @91..120
    *TARGET plant_10 100 _AGE 10
  *SOURCE plant
    *TARGET plant_10 100 _AGE 1
  *SOURCE og
    *TARGET og_10 100 _AGE 10
```

1.OUT In defining outputs it is important to include both the one-year and the ten-year period yields.

```
*OUTPUT growth
  *SOURCE ? _INVENT pi
*OUTPUT harvest
  *SOURCE cc m3
*OUTPUT inv
  *SOURCE ? _INVENT m3
*OUTPUT og_inv
  *SOURCE og _INVENT m3 + og_10 _INVENT m3
*OUTPUT sg_inv
  *SOURCE nat _INVENT m3 + nat_10 _INVENT
m3 +
    plant _INVENT m3 + plant_10 _INVENT m3
*OUTPUT all_area
  *SOURCE ? _INVENT _AREA
*OUTPUT nat_area
  *SOURCE og _INVENT _AREA + og_10 _INVENT
_AREA +
    nat _INVENT _AREA + nat_10 _INVENT
_AREA
*OUTPUT plant_area
  *SOURCE plant _INVENT _AREA + plant_10
_INVENT _AREA
```

1.OPT In the optimize section it might be necessary to include separate

```
*OBJECTIVE
  _MAX harvest 1.._LENGTH
```

	constraints for the periods covering the one-year and the ten-year periods. Remember that the increment and harvest are periodic and so might be expected to be ten times greater in the ten-year periods.	`*CONSTRAINTS`
		`  _EVEN(harvest) 1..20`
		`  _EVEN(harvest ) 21.._LENGTH`
		`  harvest[20] - .1 * harvest = 0 21`
		`  growth - harvest >= 0 35.._LENGTH`
		
		`*FORMAT MPS`

1.RPT	Nothing special here.	`*TARGET 1.txt`
		`  inv 0.._LENGTH`
		`  harvest 1.._LENGTH`
		`  growth 1.._LENGTH`

1.GRA	Nothing special here.	`*WINDOW {1} "periodic growth and harv (some 1yr some 10yr)"`
		`*WINDOW {2} "inv"`
		`*WINDOW {3} "age class"`
		`*WINDOW {4} "area"`
		`*LINES`
		`  harvest 1`
		`  growth 1`
		`  inv 2`
		`  og_inv 2`
		`  sg_inv 2`
		`  _AGECLASS ? 3`
		`  all_area 4`
		`  nat_area 4`
		`  plant_area 4`