Sequential CCSM Jan 27, 2006 meeting

Go through questions from last week: Jan 20
More questions:

- Are there more special config options that make sense? (stand-alone modes for each component?) – Stage-2 Question
  - For example: (atmlnd_only, no_lnd)
- Top level restarts for coup_csm version? – Examine with CAM group
- Timemgr namelist for coup_csm version? – ditto
  Questions for next time:
- Revist top level picture again – especially how states handled.
- List of behavior of namelist items – what you can or can not set on branch or continue?
- Error codes instead of abort?
  Look at new top level design with run methods:

```
 subroutine ccsm_seq_run( esmfsync_clock, initinfo )
 type(ESMF_Clock), intent(in) :: esmfsync_clock
 type(initinfo_t), intent(in) :: initinfo
!
! Averaging rules   components are responsible for any averaging done on within
! their coupling interval. Hub will average atmosphere fields for surface components
! and put the average on the given surface component import state. Hub will also average
! surface data to the atmosphere model coupling interval and put it on the atm import
!  state
!
  if ( shr_inputinfo_RunModel( initinfo, "atm" ) .and. ESMF_AlarmIsRinging( atm_alarm ) )then
      call ESMF_GridCompRun( gc_atm,     import=h2a_a, export=a2h_a, esmfsync_clock, phase=1 )
      call ESMF_CplCompRun ( cc_map_a2l, import=a2h_a, export=a2h_l,             phase=1, rc=rc )
      call ESMF_CplCompRun ( cc_map_a2i, import=a2h_a, export=a2h_i,             phase=1, rc=rc )
      call ESMF_CplCompRun ( cc_map_a2o, import=a2h_a, export=a2h_o,             phase=1, rc=rc )
  end if
  if ( shr_inputinfo_RunModel( initinfo, "lnd" ) .and. ESMF_AlarmIsRinging( lnd_alarm ) )then
     ! Note: x2h_l is the composite state pointing to: a2h_l
     call ESMF_GridCompRun ( gc_mrg_h2l, import=x2h_l, export=h2l_l,             phase=1, rc=rc )
     call ESMF_GridCompRun(         gc_lnd,     import=h2l_l, export=l2h_l, esmfsync_clock, phase=1 )
     call ESMF_GridCompRun(         gc_lnd,                                                 phase=2 )
     call ESMF_CplCompRun (  cc_map_a2l, import=l2h_l, export=l2h_a,             phase=2, rc=rc )
     !call ESMF_AlarmRingerOff( lnd_alarm )
  end if
  if ( shr_inputinfo_RunModel( initinfo, "ice" ) .and. ESMF_AlarmIsRinging( ice_alarm ) )then
      ! Note: x2h_i is the composite state pointing to: a2h_i, o2i_i
      call ESMF_GridCompRun ( gc_mrg_h2i, import=x2h_i, export=h2i_i,             phase=1, rc=rc )
      call ESMF_GridCompRun(  gc_ice,     import=h2i_i, export=i2h_i, esmfsync_clock, phase=1 )
      call ESMF_GridCompRun(  gc_ice,                                                 phase=2 )
      call ESMF_CplCompRun (  cc_map_a2i, import=i2h_i, export=i2h_a,             phase=2, rc=rc )
      call ESMF_CplCompRun (  cc_map_o2i, import=i2o_i, export=i2o_o,             phase=1, rc=rc )
      !call ESMF_AlarmRingerOff( ice_alarm )
  end if
  if ( shr_inputinfo_RunModel( initinfo, "ocn" ) .and. ESMF_AlarmIsRinging( ocn_alarm )  )then
      ! Note: x2h_o is the composite state pointing to: i2o_o and a2h_o
      call ESMF_GridCompRun ( gc_mrg_h2o, import=x2h_o, export=h2o_o,             phase=1, rc=rc )
      call ESMF_GridCompRun(  gc_ocn,     import=h2o_o, export=o2h_o, esmfsync_clock, phase=1 )
      call ESMF_CplCompRun(   cc_map_a2o, import=o2h_o, export=o2h_a,             phase=2 )
      call ESMF_CplCompRun (  cc_map_o2i, import=o2h_o, export=o2h_i,             phase=2, rc=rc )
      !call ESMF_AlarmRingerOff( ocn_alarm )
  end if
  if ( shr_inputinfo_runModel( initinfo, "atm" ) .and. ESMF_AlarmIsRinging( atm_alarm ) )then
     ! Note: x2h_a is the composite state pointing to: l2h_a, i2h_a, and o2h_a
     call ESMF_GridCompRun( gc_mrg_h2a, import=x2h_a, export=h2a_a, phase=1, rc=rc )
     call ESMF_GridCompRun(        gc_atm,     import=h2a_a,               phase=2 )
     !call ESMF_AlarmRingerOff( atm_alarm )
 end if
 end subroutine ccsm_seq_run
```