Task models

A task model is a definition of an instanciable task, ie

- a name
- a list of ports accepting resources from different resource types
- a list of available implementations

```
TaskModel(
        'tg',
        ports = {'output': MwmrOutput(32)} )
```

Task implementations

Software Implementation

A software implementation is defined by:

- a function to call
- a list of C source files (even if some files are shared between tasks, you must redeclare them in each task implementation)
- a list of defines to be defined later (arg `defines = {}` on task instanciation) see PerTaskCflags

```
TaskModel(
        'tg',
        ports = {'output': MwmrOutput(32)},
        impls = [ SwTask( 'tg',
                          stack_size = 4096,
                          sources = [ 'src/tg_posix.c' ],
                          defines = ['FILE_NAME'] )
               ] )
```

Hardware Implementation

An hardware implementation is in fact a dedicated hardware coprocessor (which must be declared in DSX) doing the same thing as the task. Constraint : The Task must only use MwMr channels for data exchange, as the coprocessor must use an hardware MWMR controller to interface the VCI interconnect.

Declaration consists of:

- Coprocessor reference
- Optional coprocessor arguments

```
TaskModel(
        'tg',
        ports = {'output': MwmrOutput(32)},
        impls = [ SwTask( 'tg',
                          stack_size = 4096,
                          sources = [ 'src/tg_posix.c' ] )
                 HwTask( Tg, filename = 'plan.jpg' )
               ] )
```

Virtual Hardware Implementation

If a designer wants to use an hardware implementation, but the hardware coprocessor does not exist yet, the task may be virtually implemented in hardware, which means it will use the MWMR controller to interface the VCI interconnect, and the C implementation will be used to emulate an hardware coprocessor.

```
TaskModel(
        'idct',
        ports = {'output': MwmrOutput(64),
                 'input': MwmrInput(256)},
        impls = [ SwTask( 'idct',
                          stack_size = 1024,
                          sources = [ 'src/idct.c' ] ),
                  SyntheticTask()
                ] )
```

If a task has a SyntheticTask?() declaration and an HwTask?() at the same time, the implementation chosen by DSX for simulation purposes will be unpredictable, and you should avoid such situations.