OMAC API ISSUES JOURNAL

Last Updated: June 27, 1997

June 1997 ISSUE List:
- New Action Items (see list below)
- ISSUE: What is the Naming Style, Notation?
- ISSUE: What is a Control Plan? Control Plan Unit? Examples?
- ISSUE: Object Oriented Methodology: How knowledgeable is the reader/user? How rigorous should the spec be? What industry model should be adopted?
- ISSUE: Where does the init() method register to be called at startup?
- ISSUE: Timing & Is Updatable really necessary within the API?
- ISSUE: Why are the FSM methods exposed?
- ISSUE: What happens when a method is invoked in the wrong state?
- ISSUE: How does one attach a Control Law to a Position/Velocity/Acceleration/Force Servo?
- ISSUE: What does an Object Interaction Diagram Look Like for Axis Group, Axis, Axis_Velocity_Servo and Control Law Objects Interaction?
- DISCUSSION: What is an OMAC Module?
- DISCUSSION: Are FSM, Control Plan, Capability, "OMAC Module" Modules?
- DISUSSION: Loader/Unloader Example
- DISCUSSION: How would you implement Event Service? Unresolved.
- DISCUSSION: Kinematics
- DISCUSSION: Extensibility of Axis FSM and Automating Publishing of State and Transition within Finite State Machine. Is current Axis FSM general enough?
- REVIEW: Axis
- REVIEW: Control Law

May 1997 Meeting ISSUE List:
- Reconfiguring architecture - how can it be done - Example RIGID TAP
- Error Handling, Error Propagation through the system
- QUEUES / Reset, Flush, Ultimate routes, & rewind
- Discussion on Axis -> multiple actuators
- Why is IO a MODULE and not an Infrastructure?
- Discussion on definition of Module
- Discussion on Task Coordinator running ControlPlan. What is meant by "passing"?
- Related Discussion - On tasking model
- Related Discussion - Should ControlPlanUnit be a module? => ControlPlanUnit is now a module
- HMI -discussion -separate address thread for HMI -> need proxy agent
- MMI - MMS versus GEM/SEMI specification.
- Review of Example - Reconfiguration of spindle
- Error handling and smearing of time/data through the system
- Utility of Discrete Logic Discussion
- (PLC) Computing Overload and Load Balancing Differing Views
- Timing, Synchronization, & Sequence of Operation (or Scheduling) => Scheduling or

Updater is now a module

- Example of Proxy Agent & Thread of Control
- Is cost of integration & validation of an open architecture controller is too high?
- IO Discussion
- Discussion on Control Plan Units (CPU) and machine in/dependence
- How does a ControlPlan march through its ControlPlanUnits?
- Axis Group - higher level grouping objects
- Axes Group - types of stopping, relationship to estops
- Axes Group - how to specify different acceleration profile

==

ISSUE: What is the Naming Style, Notation?

==

Example of required effort

- name_next_word() <= fcn

NameNextWord <= Class definition

nameNextWord <= class instance - object

Action => HP naming convention

Action => How do OO commercial tools do it?

See OMG, HP,

==

DISCUSSION: What is a OMAC Module?

==

def. module

- grouping of similar classes
- well-defined API
- well-defined state transitions
- replaceable by any piece of software that implements the API, state

- executable component,
- does is require a separate thread? No. (???)
- some are service related for inheritance, others are control replaceable
- is a significant piece of software in composing a controller (outside of threading)
- used to build other components, e.g., tool changer is a discrete mechanism built from OMAC modules
- no explicit wiring between modules, (i.e, no required reference architecture)

Question: Are there any wiring rules? What is the definition of a connection?

==

DISCUSSION: Are FSM, Control Plan, Capability, "OMAC Module" Modules?

==

GW: limit what is exposed by FSM to only things that are high-level states and report what state you are in. Thus, do not make FSM a separate module.

QED

-> keep control plan as module

-> add "Omac Module" as FSM module to the set of OMAC modules

-- need a new name
-- add naming, version, error logging to omac base module
-- name registry contained as static global in omac base module in system for lookup (watch out for circularities)
-> capability module - add this to the set of modules

def Capability - object to which the Task Coordinator delegates for specific modes of operation.

==
ISSUE: What is a Control Plan? Control Plan Unit? Examples?
==

What is Control Plan Unit?
Comment: distinction between Task Coordinator and CPU is fuzzy.
- Control Plan Unit could be a Task Coordinator
- Capability corresponds to traditional operating modes (AUTO, MANUAL, MDI, etc.)
  -- at Capability Level there is no coordination between Capabilities
- Control Plan Unit can be sequence of Control Plan Units
  -- example canned cycles
- can CPU be interruptible? Yes.
- Distinctions between Capability and Control Plan Unit. Could be equivalent.

Example:
- Control Plan Unit which contains tool change logic

```
class TC_CPU extends CPU
{
   void execute_CPU()
      { ag.wait_for_motion_complete();
      tool_chain.set_iopt(on);
      ... // the rest of the tool change logic
      tool_chain.set_iopt(off);
      }
   int tool_num;
   IOPT tool_chain;
   ...
   AxisGroup ag;
   TC_CPU(Axis_group, IOPT, AG, ...) { ....}

   set_tool_num(int tool)
   {
   tool_num=tool;
   };
};
```

Comments:
- Tool Changer could be Capability or CPU

Why Control Plan?
- parent of control plan units
- traversal functionality for HMI display

Revisit Issue: inverse mapping of control plan units to source code line numbers:

==

COMMENT: Problem With Documentation

==

1) distinction between TC, CPU, CP is not clear
   -- example with pseudo code would be helpful
2) Stuff on FSM in Behavior Model is poor
   -- SB: do people agree on model?
   -- SB: writeup for better understandability

==

DISUSSION: Loader/Unloader Example

==

Reference: SB Discrete Logic Example

FlowChart LUNL Example
1) Given IEC 1131 Code - logical mapping IO & functions
2) Generate number of Control Plan Units (FSM), with one associated with each state
3) Grouping of Control Plan Units become a Control Plan
4) At configuration,
   a) physical mapping IO/Fcn,
   b) Control Plan is loaded into the Discrete Logic Module
5) At initialization,
   a) set references, and
   b) register events - How? See discussion below....
6a) Clients generate events
6b) LUNL FSM execution at Discrete Logic Module scan rate

==

DISCUSSION: How would you implement Event Service? Unresolved.

==

Derived from discussion on Loader/Unloader example.

SSo:
1) clients register what events it cares about with the server capable of detecting the event
2) server send unique event id to client as part of registration
3) when server detects an event it looks in a table (linked list) of clients which care about that event and sends the event id to each client (id will be unique for each client)
4) clients use and unregister events using the id not the name.

GW:
IOPTS event subscription
- specialized iopts implements Notifier
- object to be Notified implement Notifiable

```
interface Notifier
{  void subscribe(Notifiable subscriber);
```

```
      void send_once(Notifiable subscriber);
      void set_execution_thread(Thread exec_thread);
   };

   interface Notifiable
   {  void event_notification(Notifier from);
   }
```

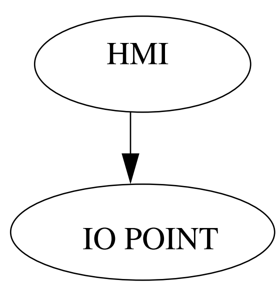
FSM events sent using methods in object

```
   class LUNL_FSM extends FSM
   {  void advance_unloader()
      {
         handle_event() ;
      }
      const EV_ADVANCE_UNLOADER=1;
      const EV_ADVANCE_LOADER=2;
      }
```

Could combine methods above
FSM subscribes to IOPT events.
event_notification() calls handle_event().

Axis Group
FSM discussion led to discussion on Axis Group finite state machine

GW: example
Fig2 CPU need not be part of system, could just be a method call to the Axis
- GW bubble figure are references, not data flow
- dispel notion of fixed reference architecture or 1 way to organize modules
- get multiple ways to organize or compose, every composition doesn't need every module. Get scalability from this principle.
Action Item: Add Simplest Architecture example:



==
DISCUSSION: Kinematics
==
Much discussion, explanation.
SSo: went of Cimetrix Tree Model of Kinematic Chains

SB: went over his Graph Model of Kinematic Structures, Kinematic Mechanisms, and Connections
Action Item: Sushil & Steve to submit some unified approach and explanation.
==

ISSUE: Object Oriented Methodology: How knowledgeable is the reader/user? How rigorous should the spec be? What industry model should be adopted?

==

Paul Johnson of Cimetrix Offered many insightful comments on the naive use of Object Oriented: technology within the OMAC document.
- Document not "traditional" OO Design Methodology or Notations. Is confusing.
- See "Design Patterns" Book for common OO design interactions. Example "Strategy Pattern"
- VISIO has OO stencils under Software Diagram
- See Figure of Paul Johnson's Comments

==

ISSUE: Where does the init() method register to be called at startup?

==

JM: could you put init() in Updatable?
GW: No, since you may want to call init() on classes that are not updatable
-- Instead, init call via Name Registry in Omac Base Class for upper most modules

==

ISSUE: Timing & Is Updatable really necessary within the API?

==

- Some OMAC modules do not have to be Updatable. Choice to run as separate thread (self-clocking, undisciplined,...)

-Example
leave processServoLoop() so that update() can call this method and possibly other pre/post methods
- can mix in threading
- SSo: Coarse & Fine Interpolation Discussion - "Growing Time" when axis limits are exceeded
-- difference timing off distance instead of time
-- going to same position, but going slower to get there
GW: may want phases within synchronous updater, 2 tasks use up 1st 2 milliseconds, 2 other tasks us up 2nd 2 milliseconds

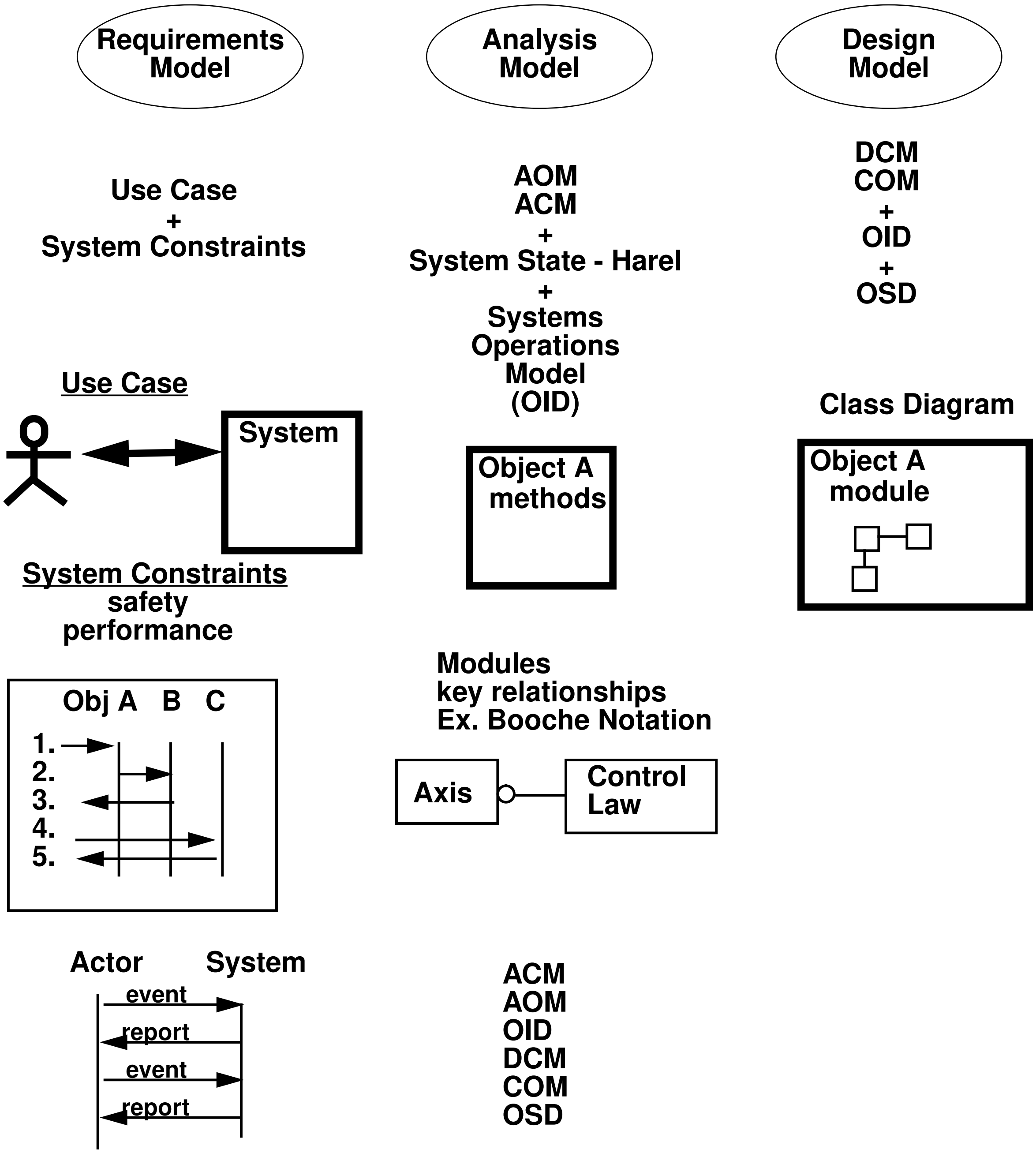
==

REVIEW: Axis

==

get_absolute_position()
-- difference between homing and zero
-- not at zero when home
-- a la relative offset
-- really redefinition of origin

Paul Johnson's Comments



QED. Took out of API.

QED. Use frames instead of datum to modify origin

axis_acceleration_servo:

Axis Group calls follow_acceleration() which sends event FOLLOW.ACCEL_EVENT to Axis,

- Axis FSM uses all other calls in the axis_acceleration_servo interface class

1. invoking follow_acceleration() causes event handling

- -- causes EVENT to be sent to FSM. Use events since multiple threads are running in the system and want to protect the queuing of the event through mutual exclusion

-- example of concurrent thread is the Task Coordinator sending Homing Events and the Axis FSM receiving the events. But HMI, or Axis Group could also send events and be in other threads.

--> some OMAC modules are aperiodic such as Task Coordinator and HMI, and can run as separate threads

--> some Control Plan Units can bypass Axis Group (such as HOMING)

==

ISSUE: Why are the FSM methods exposed?

==

So that Axis_acceleration_servo can be a replaceable component within the system. Different implementation of the component would adhere to the interface.

==

ISSUE: What happens when a method is invoked in the wrong state?

==

What is an ACCL_EVENT occurs when one is in HOMING state?

-- IGNORE? sign of weakness

-- Throw exception

==

ISSUE: How does one attach a Control Law to a Position/Velocity/Acceleration/Force Servo?

==

- Use Specialization to extend Base Class to contain Control Law component
- E.G., AxisAccelerationServo
- may not need control law component if connected to SERCOS drive

-- Base Class:

```
interface AxisAccelerationServo(){}
```

- may need control law component if doing software servoing

-- Derived Class:

```
interface CLAxisAccleationsServo() : AxisAccelerationServo
{ ControlLaw controllaw;
};
```

==

ISSUE: What does an Object Interaction Diagram Look Like for Axis Group, Axis, Axis_Velocity_Servo and Control Law Objects Interaction?

==

In lieu of a diagram, the following source code illustrates the calling sequence to facilitate following velocity servo control

```
AxGrp->fAxis->ollowing_velocity()
   Axis_velocity_servo->start_velocity_following
AxGrp->set_commanded_velocity()
```

```
Updater->Axis->update()
Axis->Axis_velocity_servo->velocity_update_action()
      Axis_velocity_servo->Axis_Command_Output->get_velocity_command()
      Axis_velocity_servo->Axis_Sensed_state->get_actual_velocity()
      Axis_velocity_servo->ControlLaw->(load parameters)
      Axis_velocity_servo->ControlLaw->calc_control_cmd
      Axis_velocity_servo->ControlLaw->(get results)
      Axis_velocity_servo->Axis_Command_Output->set_acceleration_command()
      Axis_velocity_servo->Axis_Command_Output->update()
                                    (goes to SERCOS drivers)
```

==
DISCUSSION: Extensibility of Axis FSM and Automating Publishing of State and Transition within Finite State Machine. Is current Axis FSM general enough?
==
SB: What if Software doesn't match paper?
- -- Need entire protocol for state and all state transitions
- -- Possible solutions
  - --- Add is_in_state to Axis class
  - --- add can_go_to_state(state) returns [(state, method ptr), (state, method ptr) ...]

Action Item: GW formalize FSM in automated way. Can it be done?

All: Discussion on Extensibiilty of Axis FSM
- -- is it general enough?
- - for now, can't change state or transitions, but can change actions the transitions take through specialization or component replacement
- - problem with ESTOP and different states: HARD STOP, STOP POWER, 3?, 4? that may need to be added to states and transitions

====================
AGENDA next Meeting
====================
- Axis
  - -- IO forcing
  - -- axis to multiple actuator
- Axis Group
- Process Model
- Human Machine Interface
- Review
  - Control Plan
  - Scheduling Updater
    - -- multiple inheritance issues and reduced flexibility?
    - -- event handling
    - -- watch dog timer and general timing service
  - Once and for all - IO & Event Services - bring you APIs
- Questions
  - are base classes equivalent to abstract classes

- exceptions, none enumerated, are people comfortable with try, catch, throw?

==

ISSUE: Reconfiguring architecture - Example RIGID TAP

==

SG: Configuration: 3 axes group objects created at configuration time, ag1, ag2, ag3

```
ag1: x,y,z
ag2: x,y,z, spindle
ag3: spindle
```

Problem: how do you reconfigure system to do RIGID TAP?

In the Part Program Translator, ControlPlanUnit code is generated to:
RIGID TAP = > deactivate ag1
build control plan using ag2

==

ISSUE: Error Handling, Error Propagation

==

Problem: how are errors/exceptions propagated through the system.
- Exception code treated as an event. TC gets event a and dispatches to ERROR capability.
- Events as invoking different TC capability to run a dominion Control Plan Unit to bubble up the system.
- Ex. thermal overload on drive. How does it trickle up:?=
- How to handle? QED. either add to Discrete Logic as Run or another FSM as Task Coordinator.
- synchronous versus asynchronous & solicited versus unsolicited

==

ISSUE QUEUES +
ISSUE Reset, Flush, Ultimate routes, & rewind

==

Example: Block delete to current block when considering depth of queue
- current executing block “tells” HMI that it is on
- How to handle:
  - notification of HMI/Other modules from within control plan unit
  - Not just HOLD and Resume, but stop, change tool, and then resume how does one handle queue management (as a context?)
- What can we do?
- How can we combat the high expectations of user’s perception of plug & play from the PC world?

SB: Exception handling is 90% of the problem
SB: Exception handling is 90% of the aggravation on the shop floor

==

ISSUE: Discussion on Axis -> multiple actuators

==

Example: FIXME: To Be Filled In
SS: not just an issue of multiple actuators mapping into a single axis
GW: ICON approach says to define a specialization of axis base class

Q: Should actuators be a module?
A: Now have an axis actuator class specialization.

Cimetrix finds it common enough to be built into the reference architecture
===
ISSUE: Why is IO a MODULE and not an Infrastructure?
===
Example: Consider matrix multiply. It is used everywhere. So why is IO a module?
- People want to buy/see IO.
- IO is really just a uniform front-end(API) to a device driver
(JM: what about IOCTL?)
===
ISSUE: Related Discussion on definition of Module
===
- Option 1 Modules are base classes or clustering of classes
- Option 2 Group of Responsibilities that allow proxy agent and have administrative methods
===
ISSUE: Related Discussion on tasking model:
===
SB: the TEAM philosophy was to defer tasking model to as late as possible to avoid getting hung up.
KS: lack of commitment to tasking model causes problems
KS: static library can be a module and supports init and execute, but won't really work without tasking model
===
ISSUE: Should ControlPlanUnit be a module?
===
QED: Yes.

===
ISSUE: HMI discussion -separate address thread for HMI -> need proxy agent
===
- for every object in the system you have a way to build the system
- separate address thread for HMI -> need proxy agent
- SS: "different thread to do rpc to poke/peek data"

===
ISSUE: MMI - MMS versus GEM/SEMI specification.
===
Compare.

===

ISSUE: Discussion: on TC running ControlPlan. DLL versus "passing" objects

===

- static linking of capabilities?
- smart translator would catch errors
- at static configuration -put in customize data?

How to limit translator? Translator should know xxx to limit of system done by configuration process.

Example: do a circle by small line segments, instead of line.

Heart of ISSUE: Another translator step.

1) Need for generic PPT whose output is a machine independent control plan.
2) Then a translation from generic ControlPlan Unit to a machine specific ControlPlanUnit that maps into specific object in system.

Now errors are caught at run-time.

===

ISSUE: Review of Example - Reconfiguration of spindle

===

for spindle:
- CL by axis, one velocity, on position CL
- 3 different kinds of CL being done
- hard to show how it changes between CL.
- Control Plan Unit is responsible for configuration switches.

=> rigid tap will cause change of configuration

===

ISSUE: Error handling and smearing of time/data through the system

===

SS: asked about error handling
- events as ControlPlanUnits to bubble up system.
- need more explicit Timing and Synchronization

which includes better Queueing model (Cimetrix)

Discrete Logic: Cimetrix:
- register with event notification then execute
- invoked by external subsystem

GW: sent down as ControlPlan turn on IO point in DL rung
KS: get/set any IO point

===

ISSUE: Utility of Discrete Logic Discussion

===

- People expect to see “PLC” functionality
- Will it go away with time?
- how do you handle 2000 points
- Should there be a independent agent for mist, coolant, spindle, tool?

--> How are all the independent agents coordinated if independent? How is the correctness of their performance evaluated?

- Have to fit into system performance and timing.
- scan to turn oiler on every five minutes versus oiler object should turn itself on every 5 minutes

QED: DL used to coordinate one or clusters/groups of IO points.

Compare with Cimetrix decentralized IO subscription and notification service.

- No guarantees may exceed processing power.
- Benefit of analysis may not be compare to actual testing.
- There are some bounds xxx,xxx

==

Sushil Birla: Re: Load Balancing

==

Caution against independent subscription service for event notification, independent for each module in a control system:

Problem: Incremental retrofit of event notifier xxxxx requests could unwittingly and surreptitiously corrupt the integrity of the control system due to the increase in cumulative work-load for processing and communication.

It is not possible to discover “overload” problems by “try-out” because the “overload cases” may not show up in the limited tryouts possible in practice.

The OMAC specification should include appropriate “load control” rules and constraints, e.g.,
(1) lower-priority, “if-time-available” basis for notification service
or
(2) added work-load, including communication load, completely independent of the previously existing workload, with constrained upper bound on worst-case execution-time, and no cascading dependencies and so on.

==

Steve Sorenson: Re: Load Balancing

==

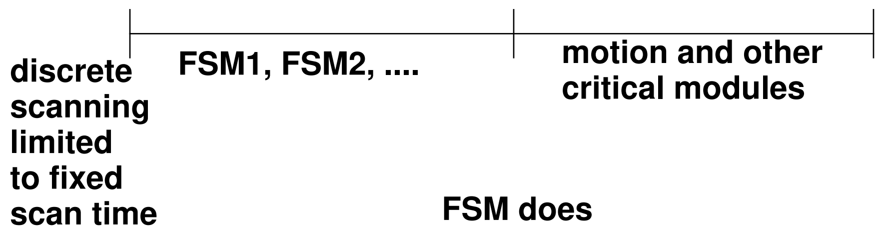
I think the integrator should have the ability to add FSM until the system runs out of processing power to deal with the extra load.

If the additional overhead causes the system to fail, this is best determined by building the system and profiling its performance. I think it is unreasonable to expect the integrator to give you his CPUP requirements and have the controller determine if it will work.

==

Ken Stoddard: Re: Load Balancing

==



FSM does
does
or does not work!?
but does not interfere with
motion and other critical modules

Requires that FSM1..n are independent
of other time critical modules

May 8th

==

ISSUE: Timing, Synchronization, & Sequence of Operation (or Scheduling)

==

KS: Platform services <= case API for scheduling
  Now- just update()
  jacking with RTOS priorities is hooey

Master Schedule
p. 17 shows Master Integrator's Process

GW: Timing Class: synchronous vs asynchronous (event)

QED: Make scheduling a Module
Platform services w/ docket scheduling : Scheduler should be a Module
- module as a cluster of object, not necessary as one that runs as one or more threads
- could have more than one updater module
- Name: scheduling Service Module
- is platform-dependent
- every module depends on service API
- module has a reference to a Updater and then registration process

```
class updatable
{void update();
]
class asych_updater
{ register_updatable();
}
```

```
class periodic_update : asynch_update
{
   get_timing_interval();
}
```

QED: Need API for Event Services - do we need Event Services
look at POSIX RT extension, look at MMS (event, and notification services), look at CORBA
key function: wait_on_event()

Is it sufficient for an Axis to block by waiting on a method through proxy agent
QED. Placeholder.

==
ISSUE: Example of Proxy Agent & Thread of Control
==
- don't know timing of Proxy Agent Technology unless you know the lower level communication mechanism
- proxy agents can run in same address space but different threads
- smart proxy agents know when you need to cross address space
- in RT system you want Proxy Agent communication mechanism exposed
- add this to the document

```
class Axis
{
FSM AxisFSM;

update() { AxisFSM.handle_evet(UPDATE_EVENT); }
home() { AxisFSM.handle_evet(HOME_EVENT); }

class FSM {
   msg_queue evq;
   int cur_state;

   handle_event(EV_no) {    evq.sent(EV_NO); }

   FSM_thread()// optional thread, this could be done in handle event
   {
      evq_receive(&ev_no);
      call_action(ev_no, cur_state);
   }

   home_update_action()
   {

   }
};
```

==
ISSUE: Beware of inherited name overloading
==

```
class event_handler()
{                  // subscribe to IO point
     execute()  ;   // instead service_event()
```

```
};

class updatable()
{
 execute() ;     // instead update();
};


class axis : event_handler, updatable
{
// handle the over temp
execute() { }

// handle the cyclic update
execute() { } // oops - name collision

};
```

==

ISSUE: Is cost of integration & validation of an open architecture controller is too high?

==

KS: Ingersoll tried but gave up, not cost effective
Can't just say here is a bucket of methods, objects and services. You could build controller if you really want to, but you probably wouldn't.

KS: OASYS had problem of small agents - easy to test for conformance, (GW: but you still have to test all the interconnections.)

Re: OASYS - need complete specification for agent - time, performance, etc.

==

ISSUE: IO DISCUSSION

==

- should an IO point tell whether its input or output
- GW: has IO derived type from IO_PT used by configuration for differentiation and type checking
- for an input discussion & tell its an input only
- expose something - for forces and simulation
- piece of data shared across domains

==

ITEM: forcing IO and machine simulations through IO points

==

- what board is this IO Pt associated?
- IO System should exist and is a container of IO points to associate IO Pt to board.
- Container is the list of IO Pts.

KS: looks in configuration file to find IO list.
SS: init build table with methods to interrogate table.
- discussion of registry to get a name

- name & handle after name lookup
- name data base

GW: name system registry - need to add a per-type basis;
table versus have it present itself (what about 2000 io points)

==

ITEM: IO meta data

==

SS: Meta data - comes out of device methods
KS: bigger point of IO than using it is, is the maintenance and diagnostics
SB: IEC1131 specifies health checking at a unit level
QED: p. 32 specifies run-time but health checking is run-time
QED:
- physical aggregation
- clients using IO
- IO themselves are shared across the system
- not just individual but as groups for efficiency

Software Class Reference - convenience IO associated with Axis Groups

Device Table
=============
Physical to Logical
Ex. 4 analog inputs, 4 analog outputs, 16 discrete io

Steve Sorenson on IO
===============
We keep a container of devices as well as a container of IO pts. Each IO point keeps a reference to a device as well as a device specific set of data which is needed to access that IO point (e.g. which bit, how wide, what type) we get the format of this data from a retrieve at start-up. This allows a configuration utility to build a GUI and supply the data which is then stored in the registry.

KS on IO:
===============
functions to:
at runtime bind device to io name
  device=> board, point, type
  this builds the internal tables
at runtime
  return handles for names for efficient access during execution

at runtime
  set and get { grouped outputs (integer)
          { single inputs

  get grouped inputs

single inputs

Steve Sorenson on device and io configuration and run-time
1. add a dll which contains which contains device interface and add to registry
2. user selects from device list in registry that he wants to configure IO on device.
3. retrieve method is called on device and GUI in built
4. user supplies data which is stored in configuration file
5. at start-up data is sent to device and reference is returned for that IO point.
6. set (an output) called.
device and IO data sent to domain interface

==

```
class IOPt
{
get_value();
set_value();
};

class IOPt_Notify : callbackNotification
{
void notify_handlers();
void attach(callbackNotification);
};

class IOPt_Notify_on_sign_change: IOPt_Notify
{

}
```

==

New: Add IO_SYSTEM as a container
-- is a list of IO points
-- meta info optional
- device reference and software class reference
- logical grouping
- distinction between polled io versus interrupt threads

New - add IO_DIRECTORY

IO Pt Attributes
name
value
monitor - null reference or object handle
bounds or get rid of units // range
(scale functions)

==

OMAC MODULE
- applies to those modules that have at least one thread of execution
- object/job has to support stop() - from discussion on OMAC_MODULE state graph
-
==

externally versus internally handled ABORT versus EXCEPTION (internally)
discussion on Shell, Program and ^C, ^Z and ^S, ^Q

==
- Question: How are parameters that are not used indicated that they aren't? Such as jerk.
- Constructor set all parameters as numbers as not used, until set
- JM: suggestion of a not a number (NaN) number approach
- KS: as opposed to as NaN, will need to be in the system as NaN-1. They have abandoned this approach.
(JM: document has max double which is NaN)
- If number set/valid, use.
=====

==
ISSUE: Discussion on Control Plan Units (CPU) and machine in/dependence
==
- add save/restore to ControlPlan
- add neutral save/restore to ControlPlan
- add transversal functions to ControlPlan

- are/can CPU save be machine independent? i.e., save_neutral You would like to add profile and then run program. This is difficult.
- CPU save can be machine independent like saving binary file.

- efficiency - lookahead is done by CPU
-- can have lookahead be done efficiently with separation
-- is a horror story with contouring
SS: process motion segment before putting on queue

SB: Action Item Re: Problem of closely spaced points, cutter compensation, backup; dynamic path modification. Take position that you can't solve all problems but don't preclude ability to get good performance.

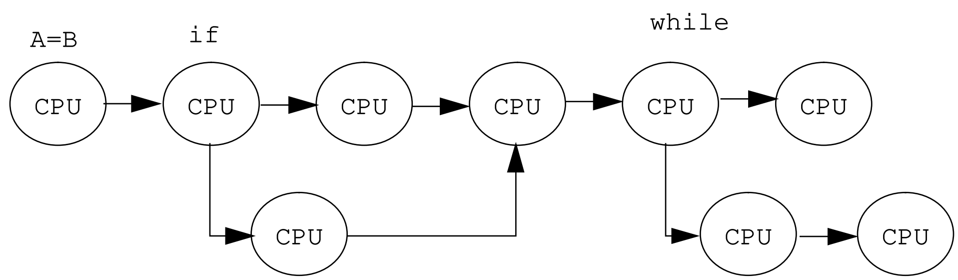
Action Item=> Agree: - Don't want to get hung up on neutral language definition.

==
ISSUE: How does a ControlPlan march through its ControlPlanUnits?
==

- What is CPU? Is it a FSM base class?

- CPU is responsible for its own branching.

```
class controlPlanUnit
{
execute_unit(); // move to control plan as
//ControlPlanUnit get_next_control
// For HMI signalling
set_active();
set_inactive();
isActive();
}
```




```
option 1:
CPU.execute()
CPU=CPU.get_next_control_plan_unit();

option 2:
CPU=CPU.execute();
NULL indicates program done
                abort?
```

Action Item=> Agree: Can't throw any CPU and entire part program must be held in memory until done. (E.G., while loop around program.) No one seemed to have any problems with this assumption.

Action Item=> Agree: fix Control Plan to be a graph, and not a tree

- KS: Save/Restore - problem of tracing back to source. Where did the CPU come from? Trellis has inverse parsable.

=> question as to whether the Control Plan (entire list) should have the execute_next method in it.

==

ISSUE: Axis Group - higher level grouping objects

==

- discussion on power management. Should be user-specifiable by ControlPlanUnit within timing. Example: set bit; wait 3 seconds; check brakes

Q: higher level objects for grouping:

-> error grouping eq. 2 live with 3 dead
-> homing by individual axis
-> power sequencing
-> power chain grouping
-> kinematic grouping

KS: naive to assume just 1 queue. Queue for motions, lookahead and blending.

==

ISSUE: Axes Group - types of stopping, relationship to estops

==

- Example - run; stop change broken tool; resume
Want to save queue context and then restore it.
- Discussion on estop and error handling modes

SS:
3 modes to stop:
: stop a max deceleration rate - hard_stop (abort)
: stop on path - pause
: stop at end of segment - hold

4 modes to recover:
: resume from current point
: skip to next segment
: flush all segments
: save/restore queue

==

ISSUE: Axes Group - how to specify different acceleration profile

==

GW: push into control plan
- replace ignore with inhibit

How does Velocity Profile Generator (VPG) fit into system at PPT level? Change to add static reference to VPG.

Can GW provide this?