Process Modelling of the TIE 97-1 Demonstration: Modelling Complex Technologies Using ACP Terminology

Stuart Aitken and Austin Tate

AIAI, University of Edinburgh, Edinburgh EH1 1HN, UK
{s.aitken,a.tate}@ed.ac.uk

11th December 1997
ISAT-AIAI/TR/6 Version 1

1 Introduction

The TIE 97-1 demonstration brings together eleven, separately developed, software systems for planning and plan evaluation. At the technological level, we can describe the software components as performing activities such as planning, scheduling, simulation and plan evaluation. Going into more detail, we can talk about hierarchical planning and Monte-Carlo methods. However, end users are more likely to conceive of the processes of Air Campaign Planning in more general, domain-related, terms such as "develop JFACC guidance" and "prioritise objectives to tasks"[2]. These gaps in terminology, and in levels of description, can be bridged by using the established ACP terminology together with IDEF3 models of the ACP process to give a context to the activities performed by the software components in the demo. We propose to use the previously elicited and verified ACP process models of Drabble, Lydiard and Tate [2] as our source of terminology and IDEF3 models.

Many tasks which occur in the wider ACP context are incorporated in the demo. Some tasks, such as the definition of the JFACC objectives, are presumed to have occurred prior to the demo. Other elements of the current ACP process are hidden, or modified, by the automation of the planning task, e.g. the processes of target selection and assessment. We argue that identifying the wider context of the demo is an important requirement for locating the new technologies in the wider application domain.

This report presents IDEF3 models which specifically describe the TIE 97-1 scenario—expressed in ACP terms. These models are explicitly represented in AIAI's process monitor tool, the purpose of which is to bridge the gap between technological and ACP views of the planning process. The relations between the states of the process products and the MPA messages that are exchanged, and Acts are also explored.

1.1 A Summary of Existing ACP Models

The ACP process model in [2] states that the planning process has five top level steps: develop JFACC guidance, develop draft support plans, develop targets, finalise support plans, and publish orders. The first and third steps are described in more detail: Develop JFACC guidance involves prioritising and sequencing objectives to tasks, and then phasing tasks; Develop targets involves four (non-primitive) processes, namely, develop candidate target list, develop draft JIPTL, provide weaponeering assessment, and develop target products. In [3] the task of identifying COGs precedes the prioritising of objectives to tasks, and hence is part of developing JFACC guidance. In [6] assigning MOMs is also grouped among these tasks.

Process Product	Process		
	develop JFACC guidance	develop support plans	develop targets
JFACC Guidance	R		R
ROE	R*		R
COG-List†	C,R		
MOMs†	C		R*
Objectives-Tasks†	C*	R*	R*
ATO			C*
Draft JIPTL			C,R,U*
Request for Guidance			C

Table 1: Processes and process products
Key: C = Creates, R = Reads, and U = Updates
(* operations that are assumed to take place; † document names to be confirmed, see Appendix 2)

These processes create, or are dependent on, particular orders, reports and letters (which will be referred to as *process products*). The following dependencies can be identified: identifying COGs requires CinC objectives (i.e. JFACC objectives) [3, 4], decomposing objectives to tasks requires COGs to be known and JFACC objectives [3], assigning MOMs requires JFACC objectives and the objective to task decomposition to be known [1]. These requirements specify a partial order on the processes of developing JFACC guidance.

Examples of the process products involved in guidance, support planning and targeting are shown in Table 1. The operations that might be performed during on these documents during a process are given in the body of the table. A more detailed breakdown of the association between subprocesses and process products can be found in the references [2, 3, 6].

The third type of model that has been elicited in the ACP domain is the verb-noun phrase model [2]. This model defines a hierarchy of verbs and states their meaning in the ACP context. These verbs were used to name the above mentioned processes in a consistent way. The full list of verbs can be found in [2], and will not be repeated here. The processes identified in the following sections of this report will also be named according to the verb model, and any exceptions will be noted.

When developing models of the TIE demos, our intention is to make maximum use of the available models and associated information. However, these models do not cover all aspects of the planning process, nor do they provide all of the detailed information we require. We shall indicate where we make assumptions—as is done above table. Such assumptions are viewed as outstanding modelling issues, which may be clarified at a later date (see Appendix 2).

2 A Process Model of the TIE 97-1 Demo

The process model of the TIE 97-1 demonstration is based on the above model of the ACP process, and the capabilities of the software components used in the demo. The modelling task is to locate the software components at the correct place in the ACP process, and to describe them in ACP terminology. Notes on process modelling issues can be found in Appendix 1.

The most relevant background material is on developing JFACC guidance. Existing models

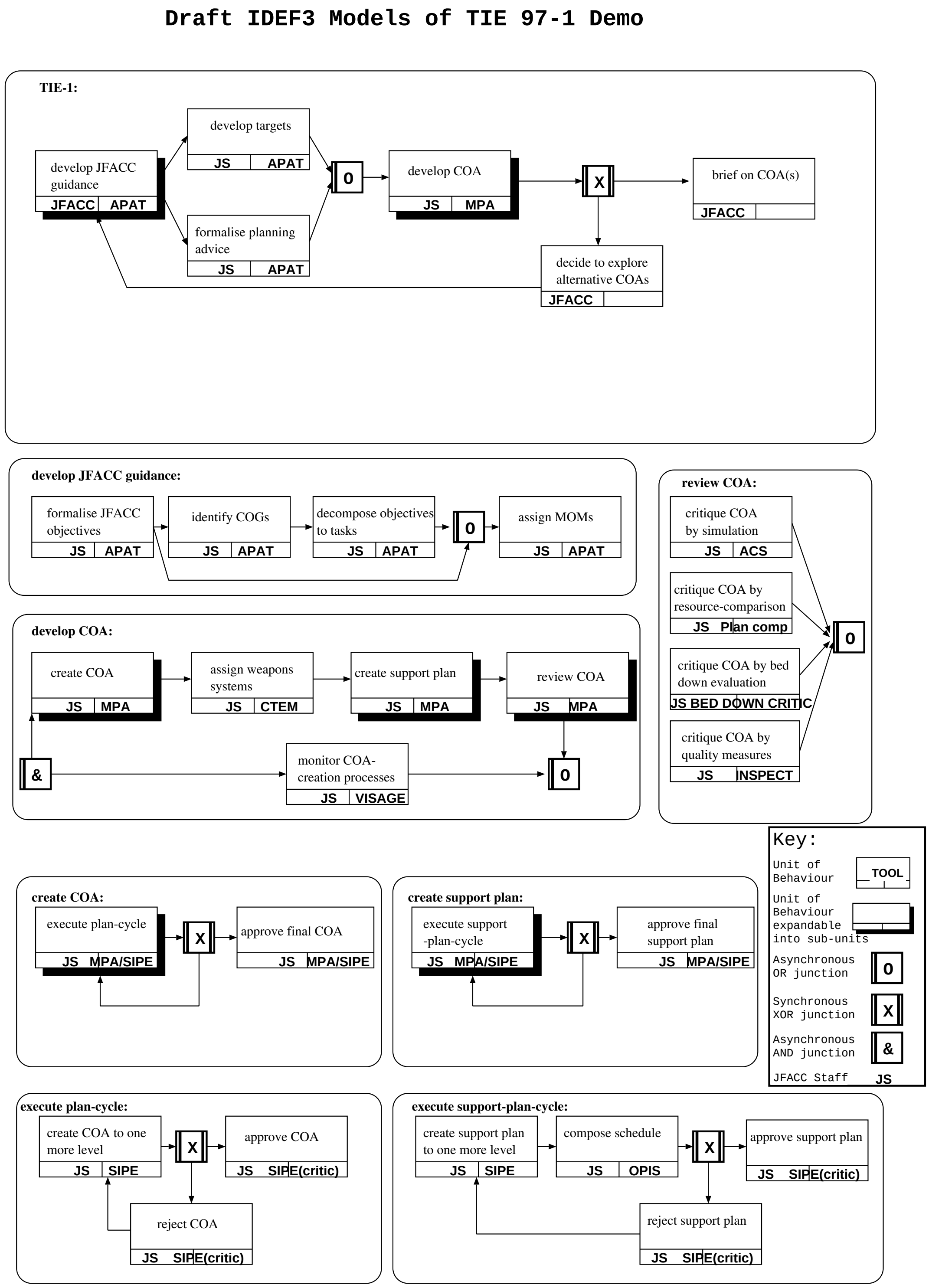

Figure 1: IDEF3 models of the TIE 97-1 demo

of support planning are insufficiently detailed for use in re-describing the planning which occurs in the demo. In contrast, detailed of models of targeting are available, but these processes are subsumed into a single planning phase in the demo. We shall therefore focus the process modelling

User	Process
CJTF	*recipient of briefing*
JFACC	develop JFACC guidance brief on COA(s) decide to explore alternative COAs
JFACC Staff	formalise JFACC objectives identify COGs decompose objectives to tasks assign MOMs develop targets formalise planning advice develop COA (and all subprocesses)

Table 2: Users and processes

effort on the processes of developing guidance and developing plans, the latter being the most interesting part of the demo.

At the top level, the following processes occur: develop JFACC guidance, develop targets, formalise planning advice, and develop COA. Formalise planning advice is a process specific to automated COA development. Advice to the automated planner is entered at this point. After the COA is developed, a decision is taken on whether to prepare a briefing on this COA or to revise the objectives (or perhaps only the planning advice) and develop more COAs. If the COA is acceptable a briefing is created then delivered. In fact, TIE 97-1 the demo starts at this point.

The development of JFACC guidance is modelled from existing materials: the objectives are formalised, COGs are identified, tasks are linked to objectives, and MOMs are assigned. We have used the verb decompose to describe the objectives to tasks link as this verb is used in the ARPI glossary [1]. (Others have used prioritise and identify.)

Developing the COA has four subprocesses for plan creation and review, and a parallel process of monitoring the planning processes. Multiple COAs can be created in parallel, each originating from different planning advice (high-level statements about what should be achieved, and how).

Two phases of planning, separated by a run of CTEM, are involved in the creation of the COA. In the first phase, SIPE finds a valid plan by deepening the initial plan level by level. CTEM is run, then the resulting combined plan is further expanded. Again the expansion is done in a level by level manner, however, in this planning phase OPIS is used to schedule resources at each level. In fact, the objectives of the planning phases are quite different: the first creates the COA, the second develops support plans. Consequently, we model these planning activities as distinct processes: create COA is followed by assign weapons systems, which is in turn followed by develop support plans.

There are four alternative ways that a COA can be reviewed. Each is characterised as critiquing, and this verb is modified by one of the qualifiers - by simulation, by resource comparison, by bed-down evaluation, or by quality measures. Each qualification corresponds to the use of a different evaluation tool to examine and form a judgement about the COA.

In step with the IDEF3 modelling method, the process model we present names the processes and defines the temporal order and logic, or branching, which is present. In Figure 1 the IDEF0 number which should appear in the unit of behaviour box has been replaced by the user name and software system name. This has been done in order to display additional information which

Software System	Process
APAT	formalise JFACC objectives identify COGs decompose objectives to tasks assign MOMs develop targets formalise planning advice
MPA	develop COA create COA execute plan cycle approve final COA create support plan execute support plan cycle approve final support plan review COA
SIPE	create COA to one more level approve COA reject COA approve final COA create support plan to one more level approve support plan reject support plan approve final support plan
Advisable Planner	formalise planning advice create COA to one more level create support plan to one more level
CTEM	assign weapons systems
OPIS	compose schedule
VISAGE	monitor COA creation processes
ACS	critique COA by simulation
Plan comparison	critique COA by resource comparison
BED-DOWN CRITIC	critique COA by bed-down evaluation
INSPECT	critique COA by quality measures

Table 3: Software systems and processes

would normally only be included in the textual description of the unit of behaviour, but which is of great interest when the process model is used as a high-level graphic of the TIE demo. These annotations are, however, non-standard.

The associations between users, tools, and the process model are given in Tables 2 and 3. These tables explicitly list the users and tools we believe to be involved in the TIE 97-1 demo, and are presented as an alternative to the process model description of the user/tool to process link. We hope these tables and figures will help the tool developers to comment on our models.

3 Comparison with Acts and KQML/MPA Traces

The Acts definitions specify a sequence of tasks that are assigned to agents (i.e. the software components). Considering the planning task, the definition of the method *pcm-expand-and-critique* states that the plan is expanded one level, then critiqued. A critic may reject the plan at this point, or may judge it to be OK. This procedure may be iterated. Once the plan is developed to the final level, and has been passed by all critics, this method is complete. The process model uses the term *create* for plan expansion, and *COA* for the plan. The actions of the critics are described as *approve* and *reject*. From the KQML/MPA message traces it is apparent that *plan-advice1*, *plan-advice2*, and *plan-advice3* are developed in this way.

The process model includes the step *compose schedule* in the planning loop for support plans as the OPIS tool is invoked to do scheduling at this point. Otherwise, the COA and support planning processes are similar. The KQML/MPA message traces indicate that *plan-advice1-phase3*, *plan-advice2-phase3*, and *plan-advice3-phase3* are developed in this way.

At a higher level, the process model is generally consistent with the decomposition of the task *generate multiple plans*. We model the *add advice* task at the top level by the process *formalise planning advice*, while it occurs in the initialisation of planning in the Acts model (i.e. at a lower level). After two or more plans have been generated they are summarised: this task is not yet reflected in the process model.

The detail of the Acts definitions is not reproduced directly in the process model, and we have also modified the terminology of the descriptions. However, we believe the process model to be a faithful abstract description of the demo. We shall continue to examine the Acts graphs of the TIE 97-1 demo using the Act-Editor, and will also further consider the traces of MPA messages. The mapping from KQML/MPA messages to process model states is specified in Appendix 3.

4 Comparison With JFACC Planning Process State Diagrams

The state transition diagrams [5] produced to describe the future JFACC planning process differ from the IDEF3 model of Figure 1 in both the modelling notation and the terminology used. The sequence of activities described in [5] is comparable with the ACP process models; the state transition model shows a higher degree of parallelisation and includes more details of the planning process. The term campaign development in [5] corresponds to develop JFACC guidance in Figure 1, force selection includes decompose objectives to tasks (a subprocess of develop JFACC guidance in Figure 1) and develop targets (a top-level process in Figure 1). Mission integration in [5] is the detailed planning process which corresponds to develop COA in Figure 1. Mission integration appears to develop strike, support and shadow activities in parallel, while in the TIE 97-1 demo strike and support planning are distinct planning phases.

5 Discussion

This report has proposed a process model of the TIE 97-1 demonstration. This model is located within the Air Campaign Planning process—as far as it has been described by previous knowledge acquisition studies. In addition to the end result, i.e. the process model itself, we have also outlined a methodology for the modelling activity, and explicitly stated the modelling issues which are currently outstanding.

References

[1] ARPI Glossary, February 1997.

[2] Drabble, B., Lydiard, T. and Tate, A. *Process Steps, Process Product, and System Capabilities.* ISAT-AIAI/TR/4, Version 2, 14 April 1997.
`http://www.aiai.ed.ac.uk/~arpi/`

[3] Kingston, J. and Lydiard, T. *Multi-Perspective Models of the Air Campaign Planning Process.* 14 March 1996.
`http://www.aiai.ed.ac.uk/~arpi/ACP-MODELS/ACP-OVERVIEW/96-MAR/HTML-DOC/acp.html`

[4] Kingston, J., Griffith, A., and Lydiard, T. Multi-Perspective Models of the Air Campaign Planning Process. *Proceedings of the 15th International Conference on Artificial Intelligence*, Pollack, M.E. ed., Morgan Kaufmann, 1997, pp. 668–673.

[5] Lafferty, L. *JFACC Planning Process State Diagrams* Softpro, Inc., 7 October 1997

[6] Lydiard, T. *Using IDEF3 to Capture the Air Campaign Planning Process.* March 1996.
`http://www.aiai.ed.ac.uk/~arpi/ACP-MODELS/ACP-OVERVIEW/96-MAR/HTML-DOC/idef3.html`

Appendix 1: Process Modelling Methodology

This report documents the method used for constructing IDEF3 models in the ACP domain. Rather than describing the components of IDEF3 models (see [6]) or a generic methodology, the aim is to describe a method for IDEF3 model construction which takes the ARPI TIE scenarios as input, and results in process models where processes are described in terms of the identified ACP verbs. The resulting models are intended to bridge the gap between descriptions of the TIE tools and technologies in technical terms, and the accepted higher-level ACP terminology.

The IDEF3 models should show the processes which result in recognised ACP process products (official documents, plans etc). The processes named in the models should be expressed in terms of the ACP verb descriptions, and, further, where a process is decomposed in to sub-processes there should be some consistency in terms of the semantics of the higher-level process and the composition of the lower level processes.

A1.1 Identifying Process Product Flow

Process modelling requires process products to be identified, and these may play the role of inputs or outputs to process steps. In the ACP domain a set of products has already been identified as the result of KA exercises [2]. A necessary part of process modelling is therefore the identification of process products (or their close equivalents) from the TIE scenarios. The operations that are performed on these products by the editing/planning tools must be known. More importantly, the goals of the users that are achieved through the use of the tools must be identified as it is the goals that are achieved (or, the tasks that are performed) that will determine the modeller's interpretation. The central task in modelling is to describe concrete user activities more abstractly as process steps — in the recognised terminology — and to show the organisation of those steps in terms of temporal and product flow dependencies.

A1.2 Ensuring Consistency

ACP verbs are organised into groups on the basis of their semantics, however, these semantics do not correspond with all possible linguistic, or intuitive meanings. The modelling process must ensure that the final models make correct use of ACP terminology (while it is likely that draft models will reflect the more intuitive meanings of the verbs). As process models are multi-level, there should also be consistency in the use of terminology between levels.

For example, consider the expansion of the process *analyse X* into the processes *choose X then publish X*. In this case, there is an inconsistency in the use of *analyse* as this verb has the notion of quantitative investigation, while *choose* has the notion of deciding but not analysing. *Publish* has the notion of issuing a document, and could be the result of an analysis process or a decision process. A replacement for *analyse X* might be *decide X* as *decide* is the verb class to which *choose* belongs (assuming that *choose* is the key activity of the *choose X then publish X* sequence).

Alternatively, further modelling may show that the process being modelled does indeed involve analysis. In this case *choose* might be replaced by a verb which captures the notion of analysis, or by a combination of processes involving analysing and choosing.

Checking the semantic consistency of process decompositions appears to be best done in a bottom-up fashion, however, modelling is likely to be both a top-down and bottom-up activity. When doing top-down modelling, top-level ACP verbs could be used in preference to sub-verbs when drafting initial ACP process models. This would allow the possibility of using more specific, and semantically consistent verbs at the expanded level. That is, the ACP verb groupings could be used as a guide to naming and decomposing process steps. In practice, process decomposition and construction is likely to be an iterative procedure. The ISI SENSUS/ACP lexicon/ontology and the **WordNet** tool could be of use in suggesting names for more specific verbs or for locating more general verbs.

A1.3 Summary

In order to construct process models we need to know:

1 the process products that are manipulated by a tool,

2 other process products that the user requires,

3 the (partial) temporal orderings of user actions,

4 the goals satisfied (or tasks performed) by tool use.

Process products should be named according to ACP terminology as we cannot deduce the proper names from descriptions. In addition, we would like to know the envisaged organisational structure within which users operate, how many users are involved, and their skill levels.

Appendix 2: Outstanding Modelling Issues

Due to the incompleteness of the information available to us the process and product models of the TIE 97-1 demo that we have developed incorporate a number of assumptions. The purpose of this appendix is to document these assumptions.

Assumptions relating to process descriptions:

- *decompose* is used as a verb in *decompose objectives to tasks* as this appears to be more standard terminology than the alternatives *identify* or *prioritise*. However, *decompose* is not an approved verb.
- *approve* is used for SIPE's critique of the COA, but *approve* also has a notion of authority which might in inappropriate to applying to an automated tool.
- qualifiers of *critique*, especially the use of *by quality measures* to describe INSPECT's function – whether this describes all of INSPECT's capabilities is unknown.

Assumptions relating to process products:

- COG-List refers to a list of COGs – whether this is the correct name is unknown.
- MOMs are associated with objectives and tasks – whether this is a distinct document or an amendment of the objectives document is unknown.
- Objective to Tasks – whether this is the correct name of the objectives document is unknown.

The process model reflects the level-by-level plan creation processes, and as yet the seven conceptual levels of COA planning (where level 1 is offensive vs. defensive, and level 2 is superiority vs. supremacy etc) cannot be mapped to plan depth. The conceptual levels of support planning are more easily mapped to plan depth, but this has not been modelled fully. The derivation of outstanding planning issues from KQML/MPA messages has not been completely determined, but appears to be possible for support planning.

Appendix 3: KQML/MPA Message Interpretation

The following tables give the process model state changes for a subset of KQML/MPA messages. Process model states are defined by the terms *begin* and *completed*. A process which *begins* will be *completed* at some future point; the beginning time may not be identifiable for some processes, therefore these processes will only be *completed*. An example of the latter is the approve/reject COA processes - only one can occur, but which one is not known at beginning of the the COA critiquing procedure. The mapping from executing/potentially executable/completed processes to display colours is not specified in these tables.

The tables do not specify that processes which follow a completed process are potentially executable - and may be coloured in the graphical display of AIAI's process monitoring tool - this information can be specified elsewhere. The begin/completion of non-primitive processes is defined in Table 1 on an 'inheritance' basis.

Note also that agent names appear as both strings and identifiers, and that the following tables were constructed from the SIPE and OPIS logs supplied by David Wilkins, SRI, Menlo Park, CA, on 1/12/97.

Message/State	Process Model/State Model Changes
`(TELL :SENDER ?pcm :REPLY-WITH ADVICE :CONTENT (:ADVICE ?s-exp :MODE ?kwd))`	formalise planning advice = completed
`(REPLY :SENDER ?planner :CONTENT (:SCHEDULE-CTEM "AS-A" ?plan ?id))`	assign weapons systems = completed
create COA to 1 more level = begin	execute plan cycle = begin
execute plan cycle = begin	create COA = begin
approve final COA = completed	create COA = completed
create support plan to 1 more level = begin	execute support plan cycle = begin
execute support plan cycle = begin	create support plan = begin
approve final support plan = completed	create support plan = completed

Table 4: KQML/MPA messages and the corresponding model state changes - higher level processes

Key: `id`: an identifier (unquoted ASCII word); `?kwd`: a keyword (i.e. `:id`); `s-exp`: an s-expression

Message	Process Model/State Model Changes

`(EVALUATE :SENDER ?pcm`
` :REPLY-WITH INIT-PROBLEM :CONTENT`
` (:INIT-PROBLEM :TASK "AS-A" :PLAN ?plan))`
[variable types: `?pcm`:id, `?plan`:string]
	`?plan` is the name of a new plan
	if `?plan` ends `-phase3` it is a support plan
	else it is a COA plan

`(EVALUATE :SENDER ?pcm`
` :REPLY-WITH EXPAND-PLAN :CONTENT`
` (:EXPAND-PLAN :TASK "AS-A"`
` :PLAN ?plan :ACTION-NETWORK ?an'))`
[`?an`:id]
	create COA to 1 more level = begin
	`?an` is the action network
	the current plan level is the final number
	in `?an` e.g. `AS-A-V1-13` indicates level 13

`(REPLY :SENDER ?planner :CONTENT`
` (:EXPAND-PLAN`
` (:LEVEL-COMPLETE "AS-A" ?plan ?an)))`
[`?planner`:string]
	create COA to 1 more level = completed
	the current plan level is the
	final number -1 in `?an`

`(INSERT :SENDER ?planner`
` :REPLY-WITH NIL :CONTENT`
` (:ANNOTATION (:PLAN-OK)`
` :TASK "AS-A" :PLAN ?plan`
` :ACTION-NETWORK ?an))`
	approve COA = completed

`(REPLY :SENDER ?planner :CONTENT`
` (:EXPAND-PLAN`
` (:PLAN-COMPLETE "AS-A" ?plan ?id )))`
	create COA to 1 more level = completed
	approve COA = completed
	plan expansion is complete

`(EVALUATE :SENDER ?pcm`
` :REPLY-WITH PLAN-OK? :CONTENT`
` (:PLAN-OK? :TASK "AS-A" :PLAN ?plan`
` :ACTION-NETWORK ?id :FINAL T))`
	approve final COA = begin
	initiate check of final COA

`(INSERT :SENDER ?planner`
` :REPLY-WITH NIL :CONTENT`
` (:ANNOTATION (:FINAL-PLAN)`
` :TASK "AS-A" :PLAN ?plan`
` :ACTION-NETWORK ?id ))`
	approve final COA = completed

Table 5: KQML/MPA messages and the corresponding model state changes for COA creation

Key: `id`: an identifier (unquoted ASCII word); `?kwd`: a keyword (i.e. `:id`)
Note: `?plan` identifies the plan throughout its creation, `?pcm` and `?planner` do not change through this demo (although in theory the `?planner` could be changed), other variables are unimportant, with the exception of `?an` where the last digits give the depth of plan expansion.

Message	Process Model/State Model Changes
`(EVALUATE :SENDER ?pcm :REPLY-WITH EXPAND-PLAN :CONTENT (:EXPAND-PLAN :TASK "AS-A" :PLAN ?splan :ACTION-NETWORK ?an))` [`?an`:id]	create support plan to 1 more level = begin `?an` is the action network the plan level is the final number in `?an` e.g. `AS-A-V1-15-POST-CTEM-1` indicates level 1
`(REPLY :SENDER ?planner :CONTENT (:EXPAND-PLAN (:LEVEL-COMPLETE "AS-A" ?splan ?an)))` [`?planner`:string]	create support plan to 1 more level = completed the plan level is the final number in `?an` -1
`(REPLY :SENDER ?planner :CONTENT (:EXPAND-PLAN (:PLAN-COMPLETE "AS-A" ?splan ?id )))`	create support plan to 1 more level = completed sipe plan expansion is complete
`(EVALUATE :SENDER ?planner :REPLY-WITH :ALLOCATE-RESOURCES :CONTENT (:ALLOCATE-RESOURCES :TASK "AS-A" :PLAN ?splan :ACTION-NETWORK ?an))`	compose schedule = begin
`(INSERT :SENDER "opis" :REPLY-WITH NIL :CONTENT (:ANNOTATION ((SCHED-COMPLETE) ?s-exp ?s-exp ?s-exp) :TASK "AS-A" :PLAN ?splan :ACTION-NETWORK ?an))`	compose schedule = completed
`(INSERT :SENDER ?planner :REPLY-WITH NIL :CONTENT (:ANNOTATION (:PLAN-OK) :TASK "AS-A" :PLAN ?splan :ACTION-NETWORK ?an))`	approve support plan = completed
`(EVALUATE :SENDER ?pcm :REPLY-WITH PLAN-OK? :CONTENT (:PLAN-OK? :TASK "AS-A" :PLAN ?splan :ACTION-NETWORK ?id :FINAL T))`	approve support plan = completed approve final support plan = begin initiate check of final support plan
`(INSERT :SENDER ?planner :REPLY-WITH NIL :CONTENT (:ANNOTATION (:FINAL-PLAN) :TASK "AS-A" :PLAN ?splan :ACTION-NETWORK ?id ))`	approve final support plan = completed

Table 6: KQML/MPA messages and the corresponding model state changes for support plan creation

Note: `?splan` identifies the support plan throughout its creation,