```
DDDD     III    RRRR
D   D     I     R   R
D   D     I     R   R
D   D     I     RRRR
D   D     I     R R
D   D     I     R  R
DDDD     III    R   R



M    M    CCC    TTTTT    III    V    V  EEEEE    SSSS            1
MM  MM   C   C     T       I     V    V  E       S               11
M M M    C         T       I     V    V  E       S                1
M    M   C         T       I     V    V  EEEE     SSS             1
M    M   C   C     T       I     V    V  E           S            1
M    M   C   C     T       I      V V   E           S     ..     1
M    M    CCC      T      III      V    EEEEE  SSSS     ..     111



*START* Job DESIGN Req #976 for EGB     Date 29-Apr-85  9:37:29 Monitor: //, TOPS
File RM:<SYSTEM.SPEC>DIR.MOTIVES.1, created:  5-Feb-85  9:41:04
        printed: 29-Apr-85  9:37:39
Job parameters: Request created:29-Apr-85  9:37:27   Page limit:261   Forms:NORMAL
File parameters: Copy: 1 of 1   Spacing:SINGLE   File format:ASCII   Print mode:ASCI
```

Benefits/costs of Evolving Directory Model as Indicated in Model.TXT.

I. Benefits.

1. Performance/Implementation.

a. Reduced and controlled connectivity. Making control points closed scopes and using library unit semantics (which are also closed scopes) reduces potential (not actual) connectivity between Ada units and makes interconnections explicit (importing through context clauses). Also, it can be arranged so that managed objects have no permanent static references. And hiding and overloading are simplified/eliminated at the directory level. This greatly simplifies the DCB problem, improving performance of change analysis, semantic analysis, file creation, etc. Also, simpler and potentially more reliable.

b. Allows more efficient directory representation. Eliminates need for overwrite operations on permanent Diana tree, although initially may be implemented that way to reduce risk.

2. Usability.

a. Consistency. Instead of three different control point kinds with very different rules for what declarations are legal, how visibility and naming works, how elaboration works, etc., there is one kind of control point with a reasonably simple model which encompasses all applications. Also unifies command semanticization context and job context.

b. Simplicity. Simple applications need not deal with full complexity of Ada. One part rather than multi-part declarative region, fewer declaration types, one-to-one correspondence between declarations and objects in control points, etc.

3. MUD Foundations.

a. Allowing program units and managed objects in subsystem simplifies maintaining subsystem state and allows arbitrary user data (documentation, etc.) to be more easily managed under the configuration policy.

b. Importing and exporting can be represented in Ada/Directory model and not through more complicated and obscure mechanisms involving configurations.

c. General simplicity makes MUD easier to design and implement (one control point kind, uniform library unit semantics, elaboration integrated into model more consistently, etc.).

d. System interface must be stabilized and released to customers at some point. New model allows simpler interfaces which can probably be stable earlier.

II. Costs.

1. Development Effort.

a. Major effort in central structures area. Partially offset by the fact that DDB had to be redesigned for performance anyway, and this model makes it easier. Also, new model is simpler. Ultimately, new model can be implemented in more reliable manner which reduces maintenance effort.

b. Code Generator (excluding support for shared elaboration) modest, but potentially destabilizing. Support for shared elaboration approximately equivalent to whatever TCP support was planned anyway.

c. Minor impact on PP, Parser and Editor.

2. General Destabilization.

This would involve destabilizing changes in two of the most crucial areas, central structures and code generation. That instability will affect higher level subsystems and will introcuce risk into A2. Currently believe risk can be factored through incremental development strategy, so other A2 functions can be released independently; however, this must be verified.