Appendix A
General usage notes

Contents

DTD tailoring

FMV does not take on any responsibility for changes discussed here, and this appendix may not be called upon to justify deviations from FMV Grund-DTD at delivery to FMV. At delivery, all files will be verified against FMV Grund-DTD as defined in its documentation.

Project definable attribute values

Use the project definable attribute values to tailor the DTD to the needs of the project. The project specific PROJDEF file should be a part of the delivery to FMV, together with descriptions of the changes, and the reasons for changes.

The changes made in PROJDEF may only be limiting in comparison to the defined values in this document (i.e. CDATA).

All project definable attribute values have comments exemplifying the possible use of the values. Any deviations to the recommended values must be reported and approved by FMV.

The exemplifying attribute values substitute the CDATA string.

Application DTD

The DTD may be slightly modified for production. The changes ought to be relatively small, e.g. changes in SGML declarations to allow minimization etc.

Apart from certain project specific needs, i.e. extra attributes etc, the DTD may need changes for production. Especially when editing is made by a team of writers, it might be wise to determine certain trails to follow down the DTD. If this is not made the large number of possible ways to tag information will probably harm aims to get documents structured in a uniformed way.

Requirements on software

Software tools that will be used for applications using FMV Grund-DTD must be set up and probably adapted (modified) in order to offer a user-friendly working environment for the user, even if the DTD has been tailored to that specific software.

Points of special interest to evaluate include:

- For the creation of links the tool must supply an interface to the HyTime codes. It is not the intention that users should key-in each code. We recommend a graphical user interface that enables the user to point and click to create a link.
- All tools must have the facility to hide elements and its content to the user. An example of this is the linking elements, which use element content as part of the addresses.

- The tool must be able to display or print attribute values, formatted as ordinary content.
- The production tool must have the facility to automatically provide attribute values for certain attributes. One example is the *id*-attribute on most elements.
- The tools must understand and handle the CALS table model described in SGML Open Exchange table model, Technical Resolution TR 9503:1995.
- Many constructs that are not coded as a table in the DTD will be presented as tables, e.g. fault-finding, materiel list, technical data. Even in a production environment it might be a good idea to capture the information in a "table like" manner. In that case, it is a requirement that the production tool can format elements side-by-side without using an ordinary table construct.