



PROGRESSIVE ADOPTION IMPROVED SYSTEM MANAGEMENT FOR OPERATIONS

DAD can be adopted for existing systems in simple user-managed stages.

Stages can overlap and some areas progress faster than others. Some areas may remain in Stage 2 for life.

In Stage 1, the existing documentation is gathered in one place. The existing files become DAD Objects and get additional properties such as a history. Frontline users can now correct document errors. Team knowledge improves as errors are corrected.

Your existing files were made by suppliers for various parts of the system. These partial views are unsuitable for the operation of the whole system. By simply reorganising the files into the way you “see” the whole system you get huge benefit from Stage 2.

Real work can be managed directly with the data. A feedback loop updates the model to the real system state.

Stage 3 implements a full SIM (System Information Model). You begin a transition to pure data. This work is far simpler in practice than you might imagine as:

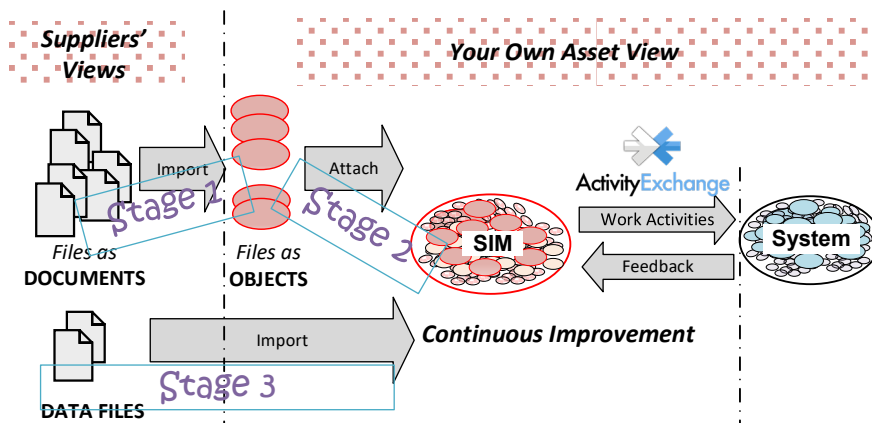
- Deleting the repeated data and ‘noise’ reduces the problem.
- Where the multiple values for the same data are imported only one is correct and that can be determined from its context in the SIM.
- Correcting the data is logical deduction applied to existing ‘evidence’.

Existing documentation separates the hardware from the software. This is crazy, it leaves individuals to guess how they act together. The SIM describes their combined functioning.

Overall, this is a practical, proven and simple “digital transition”.

Improved system knowledge

- reduces your operational risks
- reduces your dependence of outside organisations
- demonstrates conformity with requirements.





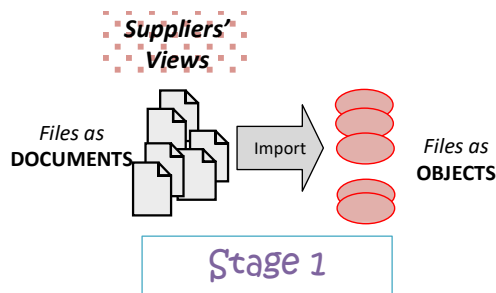
STAGE 1

Gather the system documentation files together. There will be many and these will be scattered in many places and in many formats.

Import the files into DAD.

Now you have a foundation for the SIM of the existing system.

Remove duplicate and corrupted files. Convert as many as possible to PDF. Retain the existing names of folders and files for traceability.



When imported each file becomes a DAD Object so:

- file reads and changes are logged
- files can have data appended to them.

As Objects can be grouped in DAD you can create new ways of “seeing” them. Files that appear in several places remain only one.

Users can be certain that the files they find are the latest and up-to-date.

Users can mark-up documents to record any errors they discover. These mark-ups are logged and thus controlled.

As a thought experiment, ask yourself:

- *How many places was information stored before?*
- *How long did it take users to find each file every time?*
- *Were all files accurate and up-to-date?*
- *Could you mark-up the files and ensure that new knowledge is captured?*

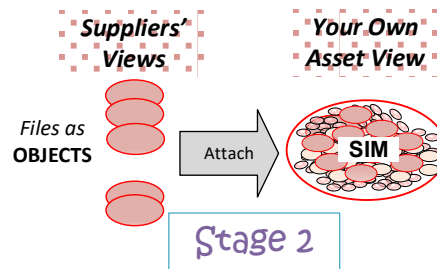


STAGE 2

Your existing files were made by suppliers for part systems. These files are arranged to allow them to deliver their parts and NOT for lifetime operation of the whole system.

By simply reorganising the files into the way you “see” the whole system you get huge benefits here in this stage.

You attach the supplier files to Objects in a SIM created as your Asset View. Now the relevant files appear to be stored in the real system itself. This is the ideal way for lifetime operation.



The links between files and objects are logged in history. As work progresses the users can modify the file-Object links to improve their relevance and they can also:

- Add data sheets, test forms and photos, etc.
- Mark-up files to correct errors or omissions.

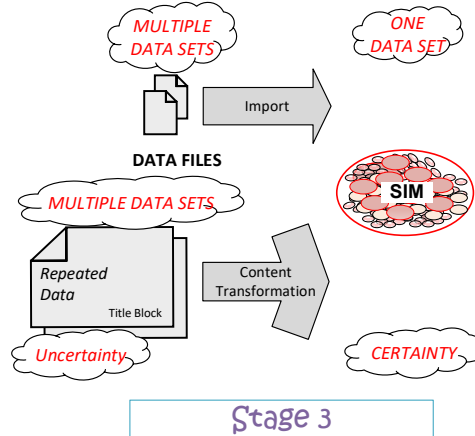
You have an platform for continuous improvement of system knowledge.



STAGE 3

This stage performs a “digital transition” on a system.

Importing data into the SIM improves knowledge and reduces risk.



There are data files for most existing systems:

- Schedules, lists, and spreadsheets
- Drawing files.
- Software configuration files

Importing schedules and lists is very low-cost and probably always worthwhile.

Drawings files are data sets. Their unique data structures mean they are impossible to import efficiently. Their data is mostly ‘noise’ and not required in the SIM. Their actual system data is frequently repetitious. Nonetheless it is often possible to copy and paste the data into DAD.

Importing software configuration files is complex and high cost. It is justified when a system is being replaced but probably not otherwise. This import requires specialists.

It is relatively simple to build the software functions by data entry. You create function blocks and signal flows for the software and link these to hardware objects resulting in a comprehensive description of the system. The behaviour of the system is now an open book for all users.

This void between hardware and software with traditional documentation has been eliminated. This work is very high value and relatively low cost. It can almost certainly be justified on risk reduction grounds as it enables:

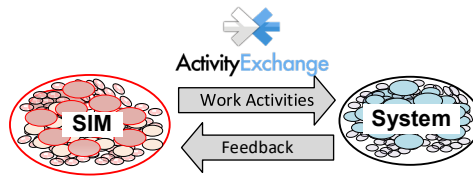
- Improved isolation management.
- Improved permit-to-work management.

A complete SIM delivers you:

- *A platform for lifetime learning.*
- *Knowledge of system hardware and functions and how these interact.*
- *A way to test conformity with requirements.*



WORKING WITH THE SIM



The SIM matches the System. Objects contain data for every part. Work instructions can use Object data. This can include external files such as Procedures and Test Sheets.

The System matches the SIM. Work activities feedback data to Objects. This may be:

- Problems/Requests for information.
- Site changes.

The history of each Object logs both data changes and work activities.

DAD has an extension ActivityExchange to:

- To separate 'site' from 'design'
- To allow cloud access on mobile devices
- For advanced project and maintenance management features.