# 4D construction learning environment development guide

**Prepared by:**

Lorinne du Toit
Marc Steffens
Rick Hill

With contributions from:

Calum Prasser
Jayesh Naran

# Table of Contents

# Overview

This guide outlines the main phases and processes involved in producing a digital application that is similar in aim and design to the AEB 4D Construction Learning Environment. The document is designed to be a resource for planning the development of projects and applications, rather than a programming manual.

Figure 1 provides an overview of the logical flow of activities and, in the leftmost column, the human resources concerned. In practice, many of these activities are more iterative than the diagram indicates.
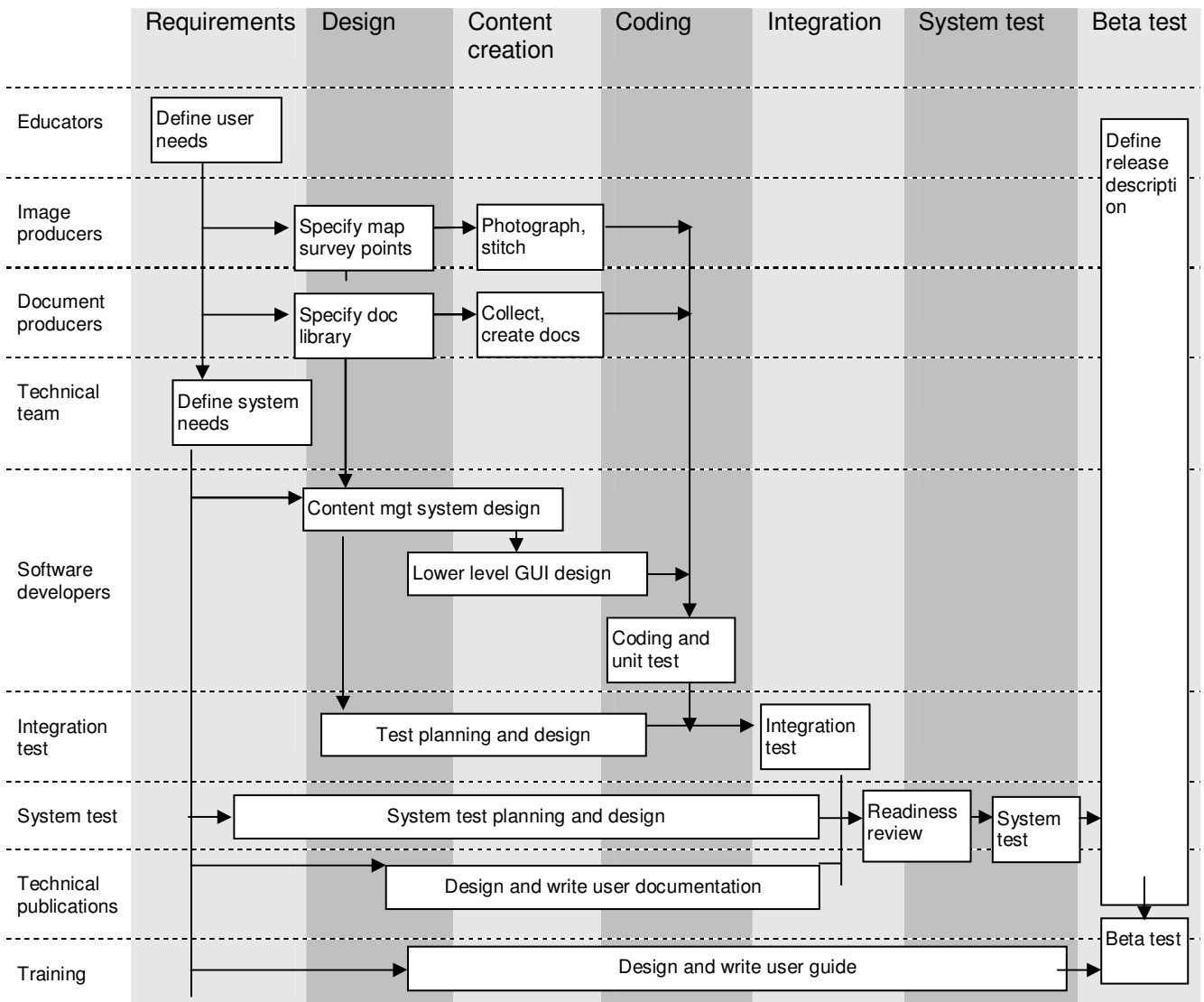


**Figure 1: Development process flow diagram**

The guide is structured according to the main phases listed in Figure 1 and content is compiled from the experience of producing the AEB 4D Construction Learning Environment.

# 1. Requirements

A software requirement specification identifies what the final product is expected to do and how it will do this (that is, its function, structure, flexibility and longevity). The specification provides the basis for estimating costs, risks and schedules as well as for the design. Requirements are commonly redefined during the development process. However, the initial specification also establishes what the software is not expected to do, and so it creates boundaries within which changes can be made without starting from scratch.

## Overall vision for the application

At the pre-planning stage, it is vital to hold a session with stakeholders to outline (brainstorm) key expectations and assumptions. This session should include broad ideas about:

- who will use the tool, when, where, for what and how (educators and students and driven by intended learning activities)
- the development time line (lead time for project establishment, the actual building construction period to be photographed, software construction and testing)
- post-production life cycle and maintenance.

## Learning activities

A reasonably detailed list of potential learning activities to be supported by the learning environment is a priority, because this will drive:

- focus and content of photography
- development of supporting resources
- functionality of the graphical user interface (GUI).

Ideally, there should be a small library of likely activities for each discipline (by program and course), noting:

- desired learning outcomes
  - theoretical content and practical skills
- types of task most likely to support this
  - individual, group work or teamwork
  - resources and tools needed for the task
- assessment strategy.

## Specification of photographic surveys

The most useful approach is to sketch out a preliminary storyboard, which can later be combined with construction site plans to map out and agree on likely camera positions.

Bear in mind that documenting all aspects of the construction process as frequently as possible can reduce the effectiveness of the virtual tour as a learning tool because:

- important elements and progress on site can become imperceptible

- navigating through the site, once the photographic data have been incorporated into the virtual tour software, will prove difficult and time-consuming.

Where learners would benefit from watching certain processes with minimal time gaps between views, consider taking video or a series of short time-lapse images, which can be linked to create a 'movie' of that process, to be used as additional resources.

**Survey focus**

Determine key areas of interest (from learning activity objectives) that will be subjects of the photographic surveys. Things to consider might include:

- ground works and retaining elements

- structure and bracing

- reinforcement design and post-tensioning layout

- concrete slab construction and service penetrations

- installation of basement and roof plant

- heating, ventilation and air conditioning systems

- wall framing and construction

- facade installation

- architectural details

- significant architectural features and spatial qualities

- finishes and landscaping

- environmental or live building aspects.

**Photographic equipment required**

The equipment listed below was used to produce the AEB 4D virtual tour imagery. This list gives basic specifications for necessary items.

Camera and lenses:

- Nikon D200 digital SLR body

- AF DX fisheye 10.5 mm f/2.8G ED fisheye lens

- AF-S DX VR Zoom-Nikkor 18–200 mm f/3.5–5.6G IF-ED digital lens for general photographic work

- AF-S DX VR Zoom-Nikkor 55–200 mm f/4–5.6G IF-ED digital lens.

Camera accessories:

- 8 GB SanDisk Extreme III Compact Card

- Nikon SB600 digital TTL (through-the-lens) flash unit for individual equipment scenes

- extra battery for camera.

Panoramic accessories:

- SLIK Master Classic tripod
- Kaidan QuickPan IV rotator base (for Nikon D200 + fisheye lens)
- Kaidan spherical bracket with quick-release camera mount
- Arca-Swiss type SLR quick-release camera plate, model PZ-109.

## Initial construction documentation needs

Prepare to collect and handle documents. Create lists of:

- topics of likely interest
- document types to be collected and/or emulated.

When this is done, you will also need to:

- obtain permissions from intellectual property rights-holders for access to and use of material such as
    - plans, drawings and specifications
    - project administration documents (reports, meeting minutes, etc.)
- make arrangements for preserving confidentiality of
    - identity for individuals and organisations represented in the material
    - 'commercial in confidence' items not covered by the assignment of intellectual property rights to the client
- arrange actual access to documentary archives (login, etc.)
- make arrangements for sign-off by document providers.

## Planning system needs

The considerations above form the basis for identifying needs in relation to:

- web and data storage hosting services
- application development and content management software
- equipment
- technical support skills (image processing, database design, graphic design, coding language, etc.).

The technical aspects listed above need to be coordinated. Despite an increasing move to open standards, hardware, software and skill sets are not always fully interoperable. So check that each proposed component will operate smoothly with the others. For example, can the application developer also manage the web server set-up? Which database software does the web hosting service support?

# 2. Design

The objective for this phase is to draft the overall production plan, identifying the resources, time line and budget. The overall technical solution, including the language, operating system, and hardware components, should be worked through and decided at this time.

Key outputs for developing the digital application are:

- a high-level (structural) design diagram (or flow chart) of the main modules, showing how the parts fit together and supported by ideas and options for appearance and functional components of the GUI
- a detailed or low-level (logical and functional) design document describing
    - database schema for content management
    - procedural steps for coding
- a quality control and testing plan, including arrangements for beta-testing.

## Mapping photographic surveys

Use the site plan, construction program and available 'for construction' building plans and drawings to refine the photographic storyboard and agree with the construction management team/site manager about the:

- likely camera positions
- frequency of surveys
- access issues to be considered.

It is important to take into account from the start that planned camera positions may become unavailable for a period or disappear altogether over time. To ensure logical and smooth navigation in the eventual virtual tour, it is vital to establish at this stage:

- naming conventions for nodes and surveys
- procedures for documenting planned and unplanned camera positions
- database structure for linking camera positions (nodes) and surveys, which will handle
    - shifts in X and Y coordinates of existing nodes
    - irregular surveys from any individual node
    - new camera positions.

## Construction documentation and other resources

The aim here is to produce a fairly detailed model of the database (schema) which will manage access to documents and other resources expected to be accessible from within the learning environment. The schema should establish:

- file naming conventions

- fields of additional information about the resource, for example, document type, status, and discipline

- search capability and terms.

The first step in this process is to draft preliminary procedures for collecting, storing and inventorying all the planned resources. Basic data needed are:

- the list of documents and other resource types to be collected and/or produced

- an indication of the generic content for each document or other resource.

The inventory design may be developed in a spreadsheet (for example, Microsoft Excel) or simple database (for example, Microsoft Access) format. This will provide the basis for the final database schema. So, aligning the file inventory and database schema from the start will reduce the workload and facilitate batch uploading of the files when the time comes.

## Graphical user interface

Expanding on the high-level design diagram, a running sheet (low-level design document) should be developed. The running sheet provides quite detailed instructions on the function and content of each page, as well as navigation and other tools to be included, both specifically for that page and more generally for all pages.

Mock-ups (images) demonstrating the running sheet instruction will help refine options for and placement of tools and other graphical elements. Errors or omissions in logic or function should become visible here and can be corrected with fewer consequences than at a later stage in the process.

For best cross-browser and mobile device support, the recommended coding approach is to use HTML5 with calls to the Jquery library. Developers can freely access large libraries of pre-scripted subroutines, objects and other elements, which can be quickly and easily embedded with appropriate alterations to specific parameters for testing.

At the end of this stage, all key decisions will have been made on the application development environment, coding language, support software, equipment and skill sets needed.

# 3. Implementation

## Content creation

Ideally, collection of documents and production of navigation maps and other resources (for example, detailed diagrams or process descriptions) should take place in parallel with the photographic surveys. This means that the designers of initial learning activities can pick up additional ideas for topics and tasks and the photographer and production team can incorporate changes to the storyboard. If this is not possible, keep a log of events and an associated list of particular documents, so that documents may be located/produced later.

**Photographic surveys for virtual tour imagery**

Brief the site construction manager and project manager about the purpose and intentions of the photo surveys and discuss the key areas of interest. Review the construction time line and note key milestone dates. The timing and scope of each survey are likely to be affected by on-site activity.

*Site access requirements*

Before access to the site is granted, the following items will need to be checked and approved:

- valid construction induction card (a White construction card in Australia)

- site induction and sign-on register

- full personal protection equipment (PPE) (construction helmet, protective glasses, gloves, high visibility long sleeve shirt, long trousers and steel-cap boots).

*Carrying out surveys*

Personnel: each photographic survey can usually be carried out by one person.

Frequency: every fortnight or when significant milestone events take place on site.

Time: the light is usually better from around 10.00 am in most time zones.

Clarify with the site manager on each visit which areas you are permitted to survey and what parts of the site are unsafe to access and survey.

Conduct a test survey to become familiar with the construction site and to check the practicality of pre-planned views.

Adjust nodes as necessary and log any changes.

Some practical tips to achieving a consistent and concise survey over the course of the construction period:

- Avoid capturing and detailing excessive and redundant construction progress and processes; consider the finished product and how it is best documented from beginning to end. For instance, documenting block wall construction at every survey node, every week, may be unproductive and time-consuming. Furthermore, it will increase the off-site image processing workload and associated file storage requirements.

- Be careful not to establish survey nodes on temporary structures as the location may become unavailable at a later point in time.

- Sometimes an obstruction at the location of a survey node is unavoidable (for example, equipment is left unattended or stacked formwork is delivered). Find the next best or closest location and identify the original node to be revisited at a later point in time. Keep a log book of survey nodes and events. Figure 2 shows an example of a log book entry.

| Date | Survey node | Floor level | Event | Solution |
|------|-------------|-------------|-------|----------|
| *02.03.2015* | *8* | *2* | *Formwork and props erected* | *Survey moved to avoid obstruction and relabeled as Node 9; survey of node* |

**Figure 2 Survey log book**

Process:

1. Prepare and test the photographic equipment.

2. Set up the camera tripod at (next) node – log any deviations to nodes in survey plan.

3. Take the photos, seven per node.

4. Copy the photos into a folder for each node.

5. Number the folders so that the number of each folder represents the order in which the node was photographed.

6. For each level, plot all the nodes for the survey on a site plan – record the X and Y coordinates of the nodes on the maps, where the zero point is the intersection between AA and A1 on the grid and the positive direction is down. See the Python Coordinates script mentioned below.

7. Renumber all the node folders using the following format: (level number).(node number) (that is, 2.4 = level 2, node 4).

**Image handling**

It is important to note that the software and processes listed here are indicative only. The specific software may not be suitable or even available for a new application development situation. Alternatives providing similar functionality can readily be substituted.

It is also worth noting that technical personnel with skills in image processing of this nature usually have a preference for, as well as greater experience in, specific software. Matching skills and software will optimise outcomes.

Software used:

- Microsoft OneNote – for digitally marking up floor or site plans with the location of survey nodes used on site

- Coordinates Program (Python script created by Jayesh Naran and Marc Steffens) – calculates node coordinates from survey locations marked up on floor plans and automatically renames folders to include the coordinates in the folder name

- AutoPano Giga V 2.6 – for stitching and rendering site photos for each survey node into large panoramic photos
- PanoTour Pro V 1.7 – for linking panoramas, inter-node and equipment hot spotting, and export of krpano panorama XML files.

Process:

1. Upload photos from survey onto the image processing PC.
2. Render in AutoPano Giga as panoramas (set JPEG quality to 12).
3. Link panoramas together in PanoTour Pro and export as XML files (set JPEG quality to 11).
4. Run the build script to package the site tour.
5. Deploy to (pre-configured) test environment.

**Minimaps**

The minimaps were drafted using computer-aided design and drafting (CADD) software:

- Ideally, use a CADD plan as an underlay to create the minimaps, for increased accuracy. Alternatively, PDF documents of plans can be scaled then traced over.
- The grid should be set out based on dimensions shown on supplied documents.
- The level of detail shown in the plan should be balanced to ensure orientation remains relative to photographs when zoomed in (expanded map) but is still easy to read when zoomed out (thumbnail).
- Export the plan drawings from the CADD program into Adobe Illustrator for final incorporation into the 4D framework.

**Construction documents**

Documents to be obtained as early as possible include:

- construction program
- drawing index
- 'for construction' site and/or floor plans.

The document wish-list, derived from the learning activity design requirements, can be used to focus the collection and categorisation of the plans, drawings, specifications, reports and other correspondence. As with the photographic surveys, it is important to be selective in order to minimise storage requirements as well as the navigation and download costs for end-users.

However, unexpected problems at the construction site or changes to building design often provide good material for learning activity scenarios, so it is worth collecting all relevant documents relating to such incidents as they occur. Documents that are not relevant to the learning activity can be removed once the issue is resolved.

The digital document library should follow the planned structure of the database schema, identifying scenario topics, categories, subcategories and major document types. Each

document added to the library should be entered into the inventory, with data entered for all fields of the schema.

Pre-designed templates for 'factional' contract administration documents (minutes, reports, requests for information, variation orders, etc.) minimise workload because relevant information can be extracted (copied) from the original and pasted into the template.

Create and keep up-to-date a list of pseudonyms for organisations and individuals involved in the construction process. Substitute these in the factional documents (using Find and replace).

### Other resources

Depending on the original specification for the environment, other resources, such as descriptive texts, animations, short videos and/or audio voice-overs, may need to be prepared in advance if these resources are intended to be linked to appropriate hotspots in the virtual tour. Hotspots are usually defined and linked to the intended resources during rendering and before exporting the (panorama) image files.

The alternative is to re-render all the surveys in a later revision, to create the hotspots and link the extra resources, before final user testing.

Resources most requested by student users in usability trials were:

- names, descriptions and/or process animations linked through hotspots in survey images
- time-lapse videos, video or audio interview clips, voice-over audio which
    - can also be included as resources from appropriate hotspots in surveys
    - need to be produced in the most accessible, cross-browser format, for example, MP4 (video) or MP3 (audio) with codecs that function for most mobile devices.

## Coding

The overall objective is to create a finished product that meets the requirements of the high-level design document. How the coding tasks are divided and sequenced depends on the specifications in the low-level design document and whether coding is carried out by one person or a team.

The development environment and application development software chosen in the design phase will influence coding requirements and time needed. Assuming that the GUI will be created using HTML5, there is a range of free and proprietary softwares that can be used. For instance, HTML5 coding can be done in a simple text editor such as Microsoft Notepad although an HTML editor with language/grammar checking greatly reduces coding time.

In the case of the AEB 4D Construction Learning Environment, all coding was performed using the Eclipse integrated development environment (IDE). This IDE, available under a free Public Licence, has editing support for all of the languages used in producing the application, including:

- HTML

- JSP

- JavaScript

- Java.

Eclipse also incorporates support for:

- an object relational mapping (ORM) framework called MyBatis, via a plugin, which simplifies the retrieval and manipulation of the data from MySQL databases

- testing via an integrated Tomcat web server.

## Integration

The integration process will be driven by the application design documents. For instance, the high-level document will steer how modules and media are integrated within the graphical user interface (GUI). As indicated in the design section, there are several options.

In the case of the AEB 4D Construction Learning Environment, the virtual tour imagery is integrated with other resources via a menu approach at user level. Basic procedures will be similar for all approaches so the integration process used in this project is provided as an example:

- Export imagery for the virtual environment in a format suitable for display using PanoTour Pro V 1.7. This creates a series of image tiles at different resolutions and an XML file for each survey that describes the details of the images required to render each node (for example, the location of each tile for each node).

- Store the XML files and images in a systematic folder structure, to which the code refers when loading each node in the web application.

- Create a MySQL database to store all the application content metadata, including survey numbers and dates, node coordinates, the path of XML configuration files used to load the photography for each node, as well as path/description/title data for the documentation library.

- Write and run a script to

    o enumerate the exported image folder structure and the XML files it contains

       and then

    o load the data into the MySQL database for use within the application.

- Use the krpano Javascript module to load the XML files for each node in response to user interaction with GUI controls in the application.

- Write and run a script to integrate the libraries of documents and other resources into the application by

    o transferring the path and other metadata for each resource into tables in the MySQL database

    o uploading the additional resources from the inventory folders.

- Use application coding to automatically build the menu structure using the data in these tables.

- Create

  - a search screen accessing the metadata in these tables (not the contents of the files themselves)

  - a password protected facility to upload additional resources on an ad hoc basis.

# 4. Testing

This aims to ensure that each element in the application operates as expected in the production environment and in tandem with all other elements (no conflicts). The application also requires quality control checking after each revision or upgrade, so a process for this should be established in the design stage and included in the project work schedule.

Testing is carried out in three separate virtual environments:

- the application development environment (for example, Eclipse), where tests are carried out on an ongoing basis by the developer as new elements are added to the overall ensemble

- the online development zone (duplicate web server), a limited access area where the main integration and quality control checks are performed by the developer and a small test team

- the production zone (web server), where the current version of the finished product is available for wider demonstration and use in learning activities.

## System tests

System testing (in the online development zone) needs to be performed after each major revision of the project. At a minimum, this should include the following steps:

1. If any images for surveys have been reprocessed or changed then view each node of the reprocessed surveys and check for errors, including image processing/stitching issues and co-ordination issues in the minimap.

2. Check each walkthrough for errors.

3. Load all document links to check for errors in the document loading process.

4. If the revision contains any new/changed functionality then the developer should perform initial testing followed by a review by other stakeholders.

In the case of the AEB 4D Construction Learning Environment application, testing was carried out manually following this protocol. However, for larger or more complex development projects it may be worth investing in automated product testing.

Once the application (or current revision) meets specification and quality control targets, it is usually deployed to the production site for beta-testing by users. Users should have a contact through which to report any previously undiscovered issues.

Any further modifications are made to the copy in the development zone, tested and then redeployed to the production site. This separation enables the developer to make and test changes with minimum disruption of service to users of an existing version.

## User evaluations and feedback

Beta-testing the application in live trials of the planned learning activities is important during the development cycle and a must within the project life. Feedback from educators and students in live tests of the activities and the tool provides opportunities not only to find hidden 'bugs' because of different expectations, devices and environments but also to make quality enhancements to learning activities and the 4D tool. Sufficient time and resources need to be available post-testing to solve bugs and refine enhancements.

A short, structured questionnaire with places for open-ended responses is a suitable instrument for gathering this information. Take care to distinguish between questions seeking feedback on the user's experience of the tool and questions about their views on the efficacy of the learning activity. An online survey, via SurveyMonkey or Blackboard, for example, is the least expensive method, particularly because basic analysis is available to download. Paper handouts in class/tutorial may encourage a greater response rate.

# 5. Documentation

Process documentation is vital to keep track of events, resources, deadlines and changes to plan during on-going development. The most appropriate approach will depend on the size of the project:

- If the development project is very large and/or entails a large production team, invest in a formal model supported by process management or document management software.

- For smaller projects/teams, an informal approach is less resource-hungry but must still provide systematic documentation that can be easily accessed for plan-tracking, formative evaluation and any audit needs.

Application documentation should be planned into the work schedule so that if an alternative technical team needs to be engaged it is possible for any new personnel to quickly become familiar with existing code and to more seamlessly continue any outstanding work plan.

User feedback provides the best guide on what to include in the application help document and how to display or describe instructions. A short animated tour of all the facilities in the 4D application may work better than a text-based document.