

# The Euclid RTP 11.13 SE Development & Exploitation Process (SEDEP)

*Keith Ford*  
Thales Training & Simulation Ltd.  
Gatwick Road, Crawley  
West Sussex, RH10 9RL  
England  
keith.ford@thales-tts.com

Keywords:

Euclid RTP 11.13, Synthetic Environments, Process, SEDEP

**ABSTRACT:** *Euclid RTP 11.13 was a major initiative to promote the use of Synthetic Environments (SEs) in Europe. One of the main results from the programme was the concept of the SE Development Environment (SEDE) for creating and utilising SEs, which is analogous to an Integrated Development Environment (IDE) for developing software applications. The purpose of the SEDE is to provide a facility that will assist the different types of SE users i.e. Problem setters, Problem Solvers, and SE Implementers, so that SEs can be delivered faster, better and cheaper. The SEDE comprises of five main components; the SE Development & Exploitation Process (SEDEP), Repository, SE Management Tool (SEMT), SE Tools (both COTS and those being developed in Euclid 11.13) and a Knowledge Base.*

*This paper describes the final version of the SEDEP, and updates the information presented at the Fall SIW in 2001. The SEDEP was developed from FEDEP v1.5 and its purpose is to provide additional information to the SE community not covered by the terms of reference of the FEDEP. In particular, it is a generic process that is not dedicated to one kind of interoperability technology and covers the complete SE lifecycle, from eliciting the user needs through to evaluating the results from operating the SE. In order to capture the work done in RTP 11.13, the FEDEP and SEDEP development teams worked together to pull-through applicable information into the IEEE 1516.3 version of the FEDEP. Following the conclusion of RTP 11.13, further development of the SEDEP has stopped whilst a new 'owner' is found for it. However, the SEDEP v2.0 is still publicly available (<http://www.euclid1113.com>) and SE developers are encouraged to use it since it complements the information provided by the FEDEP.*

## 1. Introduction

Euclid is an acronym of **E**uropean **C**o-operation for the **L**ong-term **I**n **D**efence. It provides a funding mechanism for European industry to collaborate on common defence problems and is managed by the Western European Armaments Group's (WEAG) Research Cell. Each Euclid Research and Technology Programme (RTP) is managed by a Management Group that comprises of representatives from each nation contributing funds to the programme.

The Euclid 11.13 team was led by Thales Training & Simulation (UK) and comprised of 22 European companies from 13 Nations. The programme started in November 2000 and completed in November 2003. It was the largest European Defence Research & Technology activity undertaken by the WEAG and had a budget in excess of €17 million. The aim of the project was 'to overcome the obstacles that prevent SEs being exploited in Europe by developing a process and an integrated set of prototype tools,

*which will reduce the cost and timescale of specifying, creating and utilising SEs for collective training, mission rehearsal and simulation based acquisition'.*

The aim was met by achieving the following objectives:

- Determine and mitigate obstacles that prevent networked simulations being exploited in Europe.
- Provide a process and prototype tools which will reduce the life-cycle of synthetic environment production, execution and evaluation.
- Develop software required to set-up a European repository containing information about SE assets.

An overview of the Euclid 11.13 programme is given in [1].

## 2. Terminology

In this paper, the abbreviation SE refers to ‘Synthetic Environment’ rather than ‘System Engineering’. The term ‘Synthetic Environment’ has many different meanings and there is often confusion about what is meant by an SE. Some people use the term SE to mean the instantiation of a networked simulation whereas other people take a broader view and use it to describe a framework for integrating and reusing modelling and simulation technologies to solve a range of related problems. For the purpose of Euclid 11.13, an SE was considered to be synonymous with the HLA term ‘federation’ and the supporting framework was referred to as the SE Development Environment. In the following text, the most appropriate term is used according to the context in which it is used. The term ‘SE’ is more commonly used by the ‘User’ community, whilst the term ‘Federation’ is used by the ‘Developer’ community.

## 3. SE Development Environment

One of the main deliverables from Euclid 11.13 was the development of a prototype SE Development Environment (SEDE). Figure 1 shows the conceptual framework for the SE Development Environment. It comprises of five main components:

- SE Process (known as the SEDEP) – defines the activities and tasks for creating and utilising an SE (the subject of this paper).
- SE Management Tool (SEMT) – provides capability for the management and configuration control of the SE project’s data.
- SE Tools – provides support for a specific activity or task defined by the SE process. These tools can be prototype tools developed in Euclid 11.13 or third party Commercial of the Shelf (COTS) tools.
- Repository – provides facility for exchanging data between the SE tools and for storing data so that it can be re-used on other SE projects.
- SE Knowledge Base – provides information for informing users about available SE assets and best practices for creating and utilising SEs.

Further information about the SEDE is given in [2], the SE Management Tool in [3] the repository in [4] and the SE Tools in [5].

An important aspect of the SEDE concept is that it is scalable. For small SE programmes, it can be run on a single computer or Local Area Network (LAN), whereas for large multi-national programmes, it can be distributed over many sites.

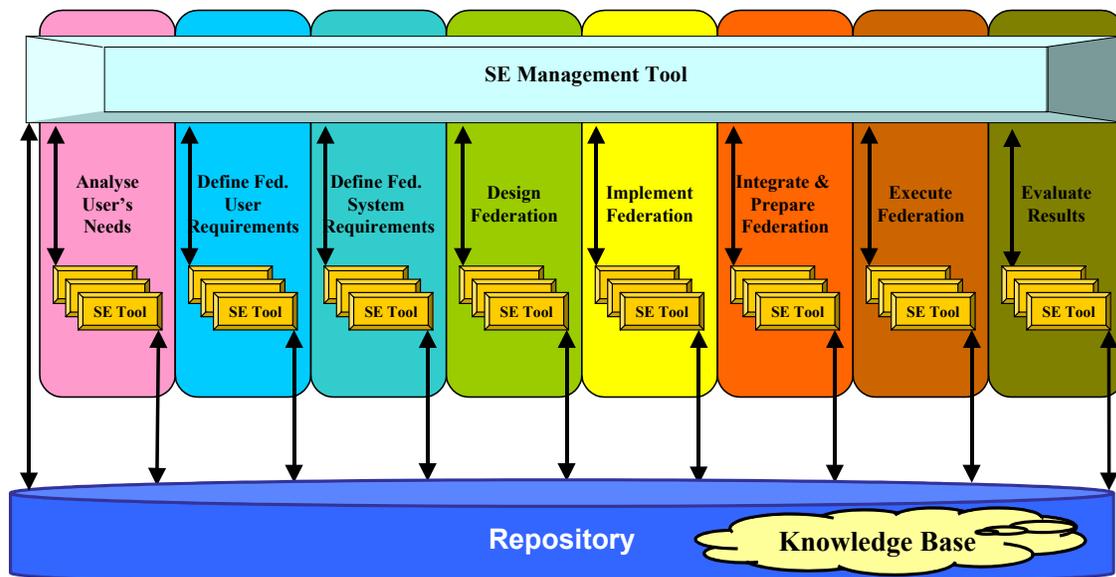


Figure 1 SEDE Conceptual Framework

## 4. SEDEP v2.0

When Euclid 11.13 first addressed the need for producing an SE process, it was decided to modify an existing one rather than developing a new process from scratch. Two options were

considered; the Federation Development and Execution Process (FEDEP), developed to support the High Level Architecture (HLA), and the TRA (Training, Rehearsal and Acquisition) process

# DRAFT

developed in Euclid 11.14. Both were published at about the same time (c.1995).

Due to the fact that the FEDEP had become a de facto standard and that there was some tool support for the process, it was decided to take the FEDEP as the baseline and to incorporate ideas from the TRA process in order to satisfy the wider needs of the SE community.

The resulting Euclid 11.13 process is known as the Synthetic Environment Development & Exploitation Process (SEDEP). The use of the term SEDEP was chosen to reinforce its close links with the FEDEP whilst promoting its more general use for developing SEs by losing its association with the HLA. A major difference between the FEDEP and the SEDEP is that the SEDEP provides the user with much more guidance by including a section on best practice and a checklist of things to do.

SEDEP v1.0 [6] was released in June 2001 and due to the amount of interest shown by the user community, was substantially revised and released as SEDEP v2.0 in November 2003.

## 4.1 Purpose

The purpose of the SEDEP is to:

- Encourage use of SE technology to benefit different application domains
- Provide guidance for developers and users to plan and perform the different activities necessary to produce the required products and results
- Promote good practice for developing SEs on time and within budget
- Promote reuse of products (federation, federates, components) and results
- Provide a framework for a tool set to reduce the cost and time for creating and utilising SEs

## 4.2 Scope

The SEDEP is relevant to all military and civil applications of SEs and covers all aspects of their specification, development and operation. It is applicable to creating and utilising small SEs, involving a few networked simulations running on a local area network, through to large SEs, running on a wide area network across national borders. Although the SEDEP uses terms from the High Level Architecture e.g. federation, the process can be tailored to support other interoperability technologies e.g. DIS. The SEDEP is most relevant where there is an emphasis on reusing existing assets and making new assets available for reuse

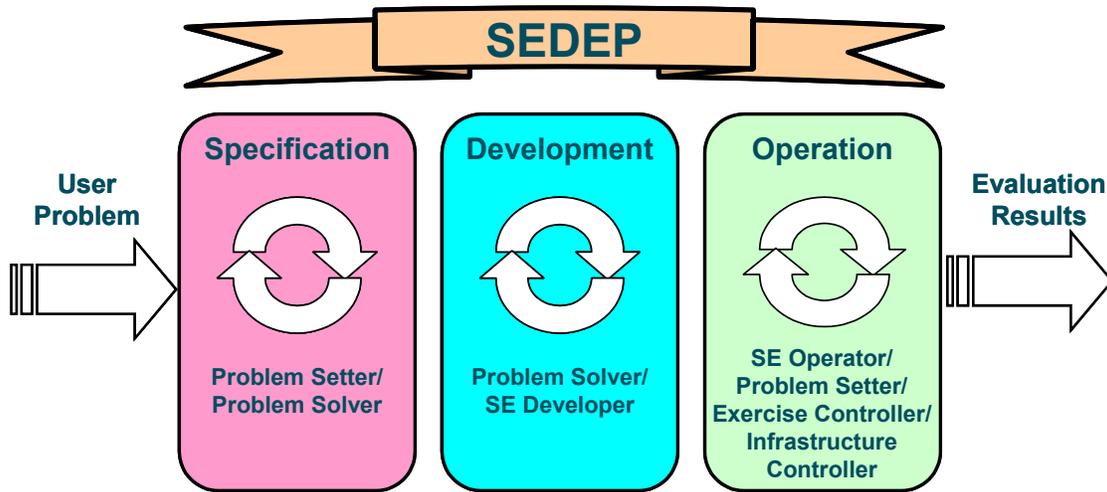
rather than the one-off standalone development of SEs.

The SEDEP is initiated when either an SE is being considered to satisfy a particular user need or when a decision has already been taken to use an SE i.e. where a through life SE master plan has been defined to support a high-level process.

The SEDEP is intended to support all the stakeholders who are involved over the lifecycle of an SE. These can be grouped into the following main roles:

- Problem setter – the individual or group that poses the question to be answered by the SE. i.e. the customer who is responsible for defining the problem and for funding the means to obtain the solution. Examples of this type of stakeholder include; End users, members of an Integrated Project Team, defence analysts, etc.
- Problem solver – the individual or group that is responsible for investigating an SE solution to the problem i.e. determines if an SE can provide a satisfactory solution and if so, defines the SE architecture and evaluation methods. This role would include the SE Systems Engineer, SE Project Manager, etc.
- SE Developers – the individual or group that develop and integrate the various elements of an SE i.e. the Specialist Engineers who provide an operational and fully tested SE. Examples of this type of stakeholder include Software Engineers, Test engineers, Database designers, etc.
- SE Operators – the individual or group that participate in the operation of the SE i.e. the people using the equipment in the SE. Examples of this role include; Operational users, Instructors, Role Players, Human Factor specialists, etc.
- Exercise Controller – the person who is responsible for overall operation of the SE at runtime i.e. for starting/stopping/pausing the exercise.
- Infrastructure Controller – the person who is responsible for the team that ensures the correct operation of the SE infrastructure whilst the SE is being operated. This role would normally be performed by the same person performing the Problem Solver role.

It should be noted that depending on the complexity of the SE, a person might fulfil more than one role when performing the SEDEP. Figure 2 shows the primary responsibilities of the stakeholders over the SEDEP lifecycle.



**Figure 2 SE Stakeholder Roles**

#### 4.3 SEDEP Relationship with FEDEP

The SEDEP is based on the Federation Development and Execution Process (FEDEP) v1.5 that was sponsored by DMSO to support the use of the High Level Architecture [6]. In order to gain the SE community's acceptance of the SEDEP, the process has been defined in such a way so as to preserve the investment in time and tools from use of the FEDEP.

The main differences between FEDEP v1.5 and the SEDEP is:

- additional step at the start of the process for analysing the user needs in order to identify existing SE assets that could be re-used and for project planning.
- additional step at the end of the process for analysing the execution outputs and evaluating the results
- generalises the wording and definition so as not to be dedicated to only using HLA technology
- specifies the use of a repository to provide a means of storing the information about the SE and for support tools to transfer data between the different process activities.

#### 5. SEDEP Overview

The SEDEP is described in a hierarchical manner so that users can delve into ever increasing detail. It is organised as Steps, Activities, Tasks, and Checklists. Two complementary representations ('views') are used to describe the SEDEP:

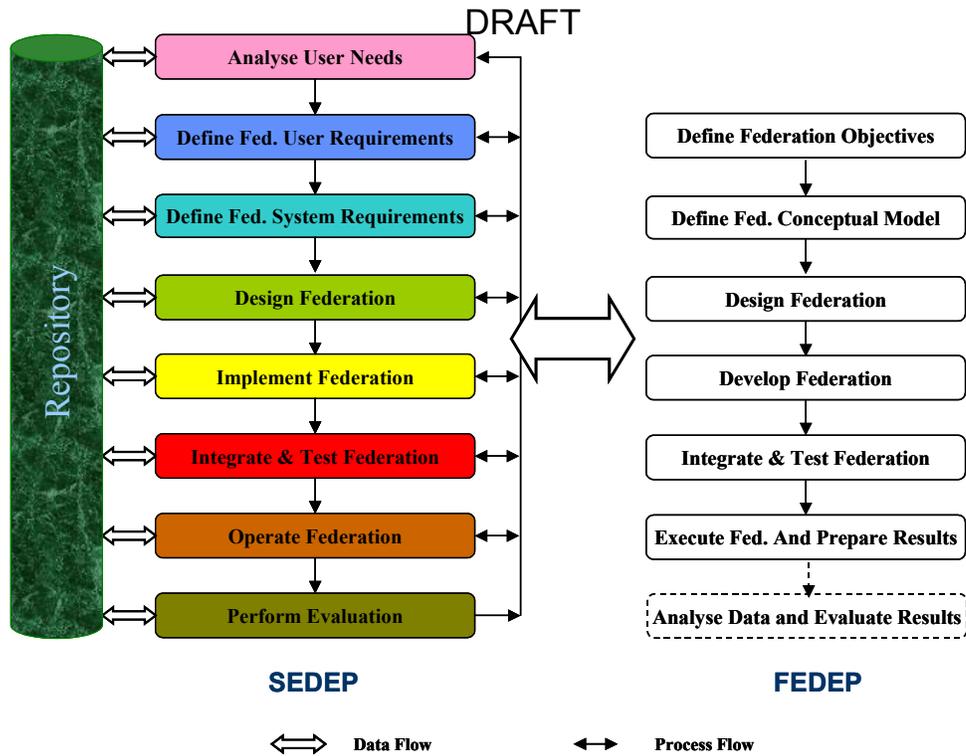
- **Steps representation:** the different activities of the process are organised in steps sequentially along the whole process. Each step covers a specific phase of the SE lifecycle.
- **Overlays representation:** the different activities of the process are thematically organised in overlays along the whole process. Each overlay covers a specific theme or aspect of the SE, which requires a specific technical environment and/or specific expertise.

#### 5.1 Steps Representation

The SEDEP is defined by 8 steps and each step comprises of a number of activities. Each step has its own feedback loops and multiple internal iterations may be performed without interfacing to other steps. The interfaces between steps are more formal and occur less frequently than between activities within a step.

The activities are defined according to the following principles:

- can or should be performed by (or under the authority of) one person.
- can or should be performed at one time, or do not need to be separated in time
- can or should be performed in one location
- are likely to be supported, or need to be supported by the same tool
- take input from a single (or closely related) identifiable set of data
- contribute to a single (or closely related) set of output data



**Figure 3 SEDEP Steps**

The 8 steps of the SEDEP are shown in Figure 3. These are numbered from 0 to 7 so as to preserve the mapping with the FEDEP steps. The process is underpinned by a Repository, which serves two purposes. To store information about existing SE assets e.g. simulators, databases, etc (to promote re-use), and to store data generated by an activity so that it is available as input to other activities. Although the archiving of SEDEP products is not explicitly stated in the SEDEP description, it is assumed that this is performed throughout the process.

An important aspect of the SEDEP is that it may be used iteratively. This means that it may be initiated several times for a particular SE project and that successive iterations build on the information already available. The SEDEP is tailored for each iteration to meet the requirements of the objective of the iteration. This means that for some iterations, some activities are performed to a low level of detail or even not performed at all. An example of tailoring the SEDEP on an SE project is the specification of the particular tools used to support the different activities.

Although the steps are shown as a sequential set of operations, in practice due to demanding timescales, it may be necessary to start an activity before the preceding one has been completed. This can be achieved by producing an initial output and iteratively refining it until the final version is obtained. There are also feedback loops where it

may be necessary to revisit an earlier step as a result of actions performed in later ones.

- **Step 0: Analyse User Needs** The purpose of this step is to define and analyse the users needs in order to understand what results the SE should provide and the purpose of the current SEDEP iteration. This information enables the Problem Solvers to plan how the SEDEP iteration should be performed. The project planning not only includes traditional project management planning activities but also tailoring the process to satisfy the requirements of the SEDEP iteration. An important outcome of this step is to reduce the cost and risk of producing the SE by capitalising on the experience from previous SE projects and to identify suitable SE assets or solutions that could be re-used.
- **Step 1: Define Federation User Requirements** The purpose of this step is to provide a comprehensive description of what the Problem Setter(s) wants from the Federation. This is achieved by the Problem Setter and Problem Solver working together to define the Federation User Requirements, the Scenario to be performed and the Evaluation objectives. The Evaluation Objectives provide one of the main drivers as to what capability the federation should provide and it is essential that it be considered at this early stage in the process.

- **Step 2: Define Federation System Requirements** The purpose of this step is to define the specification for a federation that will satisfy the User Requirements. This step is mainly performed by the Problem Solver. The derivation of the Federation System Requirements is performed in three phases. First of all the User Requirements are transformed into an implementation independent conceptual model of the federation. This not only provides a convenient vehicle for describing the operation of the federation but it also provides a convenient way of documenting it so that it can be reused. Secondly, the information provided by the User Requirements and conceptual model is transformed into a specific set of testable system requirements for the federation. Thirdly the Evaluation Requirements are developed, which specifies the evaluation criteria, methods, algorithms etc. to satisfy the Evaluation Objectives.
- **Step 3: Design Federation** The purpose of this step is to produce the design of the federation that will be implemented in Step 4. This involves identifying existing federation elements that can be re-used, either with or without modification, performing the high-level federation design and the detailed design of the elements that comprise the federation. Federation elements include simulations, simulators, databases, data loggers, network infrastructure etc. In parallel with the design activities, the federation plans are produced for integrating and testing the federation.
- **Step 4: Implement Federation** The purpose of this step is to implement and unit test all aspects of the federation design produced in the preceding step. This includes implementing the network infrastructure, federates, databases, evaluation algorithms etc. Depending on what's already available, either new federation elements are implemented from scratch or existing ones modified.
- **Step 5: Integrate & Test Federation** The purpose of this step is to configure and integrate the Federation elements with the runtime infrastructure and test the Federation to ensure that it is ready for operation. This includes testing the interactions between Federates, ensuring that the network is reliable and can handle the expected traffic, verifying the Federation against the system requirements, and validating the Federation against the user requirements.
- **Step 6: Operate Federation** The purpose of this step is to prepare the federation for operation, run the federation scenario(s) and to

collect data for performing the evaluation. Preparation activities may include training the instructors, operators, and technicians participating in the federation and to rehearse the federation executions to identify unforeseen problems. Depending on the federation requirements, it may be necessary to perform several federation executions either using the same scenario or with a modified scenario.

- **Step 7: Perform Evaluation** The purpose of this step is to analyse the data acquired during the federation execution and evaluate the results. These are fed back to the Problem Setter to decide if the problem being investigated has been solved or whether further work is required. In the latter case, it may be necessary to perform another iteration of the SEDEP based on the recommendations from the current iteration. Another aspect of evaluation is to evaluate how well the federation solved the problem and whether any improvements could be made. This information forms part of the SEDEP knowledgebase and can be used to optimise the development and execution of other federations.

The description of the SEDEP steps comprises of the following topics:

- Overview of Step: Describes purpose of step and main issues.
- Data Flow Diagram: Portrays main data flows between activities. However, it should also be noted that additional data flows might exist especially after tailoring.
- Library descriptions: Describes content of library and the tool typically used to create the data.
- Activity descriptions: Describes purpose of activity and main issues.
- List of Tasks: Lists main tasks associated with an activity.
- Support Tools: Provides example of a tool used to support an activity.
- Best Practice: Provides feedback of practical experience about performing an activity.
- Checklist: Describes list of things that could be done. This will be tailored to meet the requirements of a given federation.

Table 1 shows the activities associated with each SEDEP Step.

Analyse User Needs	Define Federation User Requirements	Define Federation System Requirements	Design Federation	Implement Federation	Integrate & Test Federation	Operate Federation	Perform Evaluation
Analyse User Inputs	Develop Federation User Requirements	Develop Federation Conceptual Model	Design Federation	Prepare Federation Infrastructure	Configure & Test Federation Infrastructure	Produce Operation Plan	Analyse Data
Search Existing SE Data	Develop Evaluation Objectives	Define Evaluation	Design Federation Elements	Implement Federation Elements	Integrate & Test Federates	Prepare Federation for Execution	Evaluate Analysed Data
Project Planning	Develop Scenario	Develop Federation System Requirements	Produce Integration & Test Plan	Define Federation Data	Test Federation	Execute Federation	Evaluate Federation
			Design Evaluation	Implement Evaluation		Prepare Federation Data	
						Collect Supplementary Evaluation Data	

Table 1 SEDEP Activities

**5.2 Overlay Representation**

The purpose of the overlays is to provide more specific information to assist the SEDEP user. Two types of overlay have been defined; those based on user roles e.g. Problem Setter overlay, and those based on particular areas of SE development and use e.g. Evaluation overlay. A third type of overlay is envisaged for future versions of the SEDEP, that is, application overlays. These would give specific help on how the SEDEP should be used to support different application domains e.g. Acquisition overlay. SEDEP v2.0 describes six overlays. These are:

- Problem Setter Overlay – describes the responsibility of the Problem Setter for each SEDEP step.
- Problem Solver Overlay – describes the responsibility of the Problem Solver for each SEDEP step.
- SE Developer Overlay – describes the responsibility of the SE Developer for each SEDEP step.
- SE Operator Overlay – describes the responsibility of the group of people that will use the SE e.g. instructors, role players, etc. for each SEDEP step.

- Verification & Validation Overlay – describes what V&V tasks need to be performed for each SEDEP step.
- Evaluation Overlay – describes what evaluation tasks need to be performed for each SEDEP step.

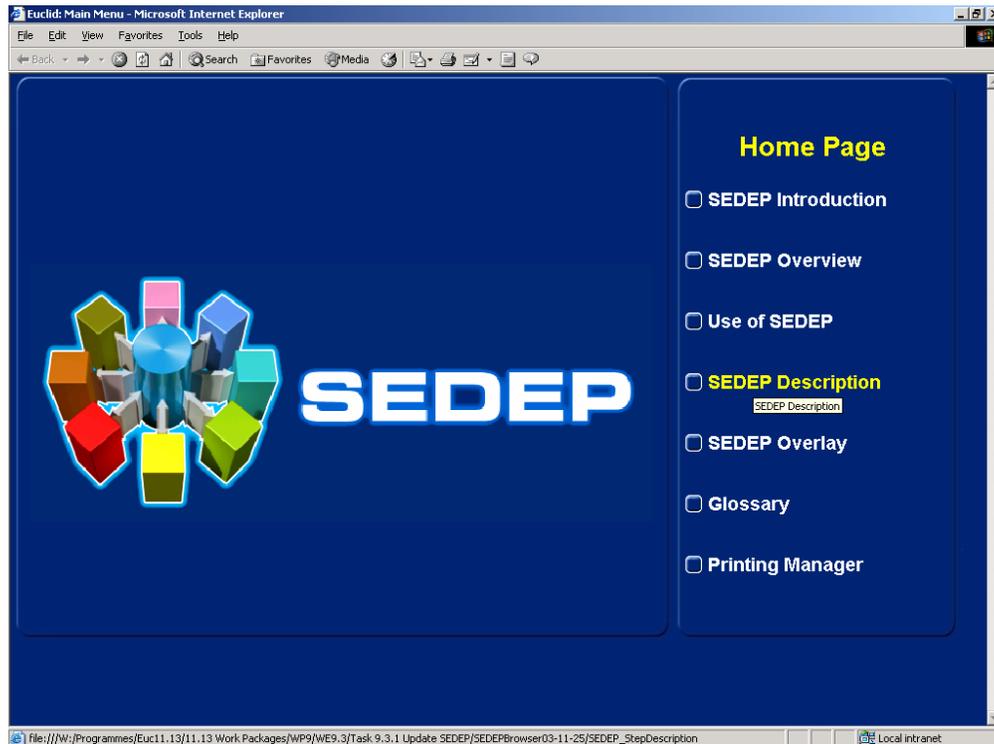
It is intended that additional overlays are added to future versions of the SEDEP. These might include application domain overlays, an SE specification overlay, project management overlay etc. It is hoped that future versions of the SEDEP overlays will draw upon the results from the Software Engineering Institute (Carnegie Mellon) work on Capability Maturity Models (CMMs).

**5.3 SEDEP Browser**

An innovative feature of the SEDEP is that it is available in electronic form as the SEDEP browser. The SEDEP description is stored as html pages and can be viewed with a standard Internet browser. It also means that the information can be made available over the Internet and is available from the Euclid 11.13 web site (<http://www.euclid1113.com>). Another advantage of providing the process on-line, is that the information provided in the ‘Best Practice’ and ‘Checklist’ sections can be updated much more easily. It is intended that as people use the process,

# DRAFT

their experience is captured, vetted and made available to others.



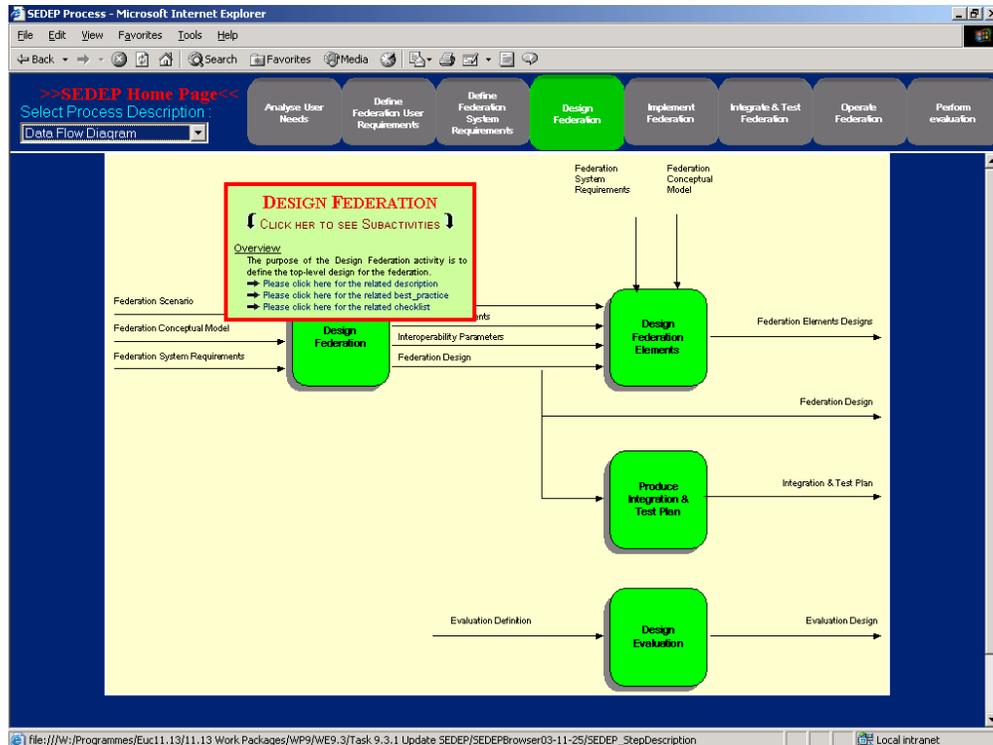
**Figure 4 SEDEP Browser ‘Home Page’**

The ‘Home Page’ of the SEDEP Browser is shown in Figure 4. This provides the user with access to high-level information about the process (SEDEP Overview), detailed description of the process (SEDEP Description), description of the SEDEP overlays, and the ability to print either all or part of the process (Printing Manager). To simplify updating the process, all the information about the SEDEP is held in a database and a ‘Browser Generator’ has been produced for automatically creating the html pages.

For each step of the process, the SEDEP description provides:

- Overview of step
- Data Flow Diagram
- Recommended Tools
- Checklist
- Best Practice
- Activity Descriptions
- Input/output descriptions

The information can be obtained from the drop down list box (top-left of diagram) or by placing the cursor over an activity on the data flow diagram (Figure 5).



**Figure 5 Example of SEDEP Browser Data Flow Diagram**

## 6. Conclusions

The SEDEP has been one of the most visible, and unexpected, outcomes from Euclid 11.13. A lot of interest has been shown in having an 'SE' process that is not dedicated to a particular interoperability technology and the MoDs from several countries in Europe e.g. U.K and France, are evaluating its use. The SEDEP is currently available from the Euclid 11.13 web site (<http://www.euclid1113.com>).

Since the previous SEDEP paper was presented at the 2001 Fall SIW, the process has undergone a major revision. This has been as a result of experience of using it on a project run as part of Euclid 11.13 known as Nordic Fox. This was a 'fictitious' project where it was assumed there was a need to build an SE to investigate issues associated with updating an Air Defence System. Although an executable federation was not implemented, all the documentation was generated as if one were to be produced and issues about the usefulness and completeness of the process fed back prior to the SEDEP's update to v2.0 [9].

Initially it was thought that the FEDEP and SEDEP should merge by taking the best aspects from each process. To facilitate this, the SEDEP team participated in the review of the FEDEP before it

became an IEEE standard. As a result, the latest version of the FEDEP i.e. IEEE 1516.3 [8], has incorporated some of the ideas promoted by the SEDEP v1.0 (in particular, the definition of a step dedicated to analysing and evaluating the results from the SE). However, it is now thought that since the process is aimed at a wider audience than the FEDEP, maybe it should exist in some form in its own right. Options for the future of the SEDEP include incorporating it as an 'overlay' to the FEDEP, maintaining it as a complementary process to the FEDEP, or extending the scope of the FEDEP by merging the two processes when the FEDEP comes up for review.

The main challenge now is to find a champion for the SEDEP since now that Euclid 11.13 has finished, no organisation has currently been identified to take responsibility for it. Rather than individual nations developing the SEDEP and promoting it for their own needs, it would be desirable that its exploitation is co-ordinated by a single internationally respected organisation. Maybe this is a role that SISO could fulfil, as support of the SEDEP by SISO would also help ensure that the information provided by the SEDEP and FEDEP is seen as being complementary rather than contradictory.

## 7. Acknowledgements

The author would like to acknowledge the help of all his colleagues in the Euclid consortium with the development of the SEDEP, particularly Pascal Peyronnet (Thales Training & Simulation) who was the inspiration for the initial work.

## 8. References

- [1] K. J. Ford, Euclid RTP11.13 Realising the Potential of Networked Simulation in Europe European SIW, June 01, Paper 01E-SIW-011.
- [2] K.J. Ford, The Euclid RTP11.13 SE Development Environment, Euro SIW June 02, Paper 02E-SIW-007
- [3] K.J. Ford, Marko Jokipii, The Euclid 11.13 SE Management Tool, Euro SIW June 03, Paper No. 03E-SIW-093.
- [4] Marco Fabbri, Rui Nunes, Roger Davies, Erik De Koster, The Repository in support of the Synthetic Environment Development & Exploitation Process (SEDEP), ITEC 2003.
- [5] K.J. Ford, The Euclid RTP11.13 Tool Suite, ITEC Paper No. 49, April 2003.
- [6] K. J. Ford, Pascal Peyronnet, The Euclid RTP 11.13 Synthetic Environment Development & Exploitation Process, Fall SIW, September 01, Paper 01F-SIW-124.
- [7] High Level Architecture Federation Development and Execution Process (FEDEP) Model, Version 1.5, December 8 1999.
- [8] IEEE Standard 1516.3, IEEE Recommended Practice for High Level Architecture (HLA) Federation Development & Execution Process (FEDEP), 23 April 2003.
- [9] Goran Ancker, The Euclid RTP 11.13 SE Development & Exploitation Process (SEDEP): Practical Experience from Nordic Fox Use Case, Spring SIW 2004, Paper No. 04S-SIW-066.