



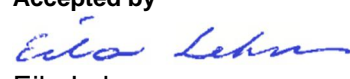


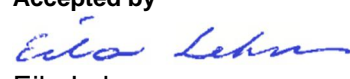


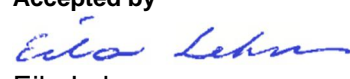
**RESEARCH REPORT**

VTT-R-00257-16

# **A draft architecture for a level 3 PSA code**

Authors: Ilkka Karanta

Confidentiality: Public

<b>Report's title</b> A draft architecture for a level 3 PSA code				
<b>Customer, contact person, address</b> VYR	<b>Order reference</b> SAFIR 4/2015			
<b>Project name</b> Probabilistic risk assessment method development and applications	<b>Project number/Short name</b> 101958/PRAMEA			
<b>Author(s)</b> Ilkka Karanta	<b>Pages</b> 18/			
<b>Keywords</b> level 3 probabilistic risk analysis, software development	<b>Report identification code</b> VTT-R-00257-16			
<b>Summary</b> <p>An initial sketch of a software architecture for code conducting consequence assessment of severe nuclear accidents (level 3 probabilistic risk analysis) is presented. The code, if implemented, would complement FinPSA Level 1 and FinPSA Level 2 codes being developed by VTT. The purpose of the sketch is to facilitate communication with stakeholders (NPP companies, regulatory agencies, developers), and form a basis to further architectural specification and software development if a decision is made to develop the software.</p> <p>Existing applicable software is reviewed, with emphasis on free and open source software. Good alternatives for conducting atmospheric dispersion computations, geographic information management, and statistical and mathematical analyses exist. Also some code directly applicable to level 3 purposes is available.</p> <p>The functional and information viewpoints of the architecture are presented. In the functional viewpoint, the main components, their interfaces, and dependences between them are outlined. The code shall include components for atmospheric and aquatic dispersion estimation, health consequences assessment, and economic consequences assessment, among others. In the information viewpoint, the most important (in terms of data to be handled) information is presented.</p> <p>Existing functional requirements are mapped to components taking care of their implementation.</p>				
<b>Confidentiality</b>	Public			
Espoo 11.2.2016 <table border="0" style="width: 100%;"> <tr> <td style="width: 33%; vertical-align: top;"> <b>Written by</b>              Ilkka Karanta            Senior Scientist         </td> <td style="width: 33%; vertical-align: top;"> <b>Reviewed by</b>              Kim Björkman            Research Scientist         </td> <td style="width: 33%; vertical-align: top;"> <b>Accepted by</b>              Eila Lehmus            Head of Research Area         </td> </tr> </table>		<b>Written by</b>  Ilkka Karanta Senior Scientist	<b>Reviewed by</b>  Kim Björkman Research Scientist	<b>Accepted by</b>  Eila Lehmus Head of Research Area
<b>Written by</b>  Ilkka Karanta Senior Scientist	<b>Reviewed by</b>  Kim Björkman Research Scientist	<b>Accepted by</b>  Eila Lehmus Head of Research Area		
<b>VTT's contact address</b> Ilkka Karanta, Box 1000, FI-02044 VTT, Finland, ilkka.karanta@vtt.fi				
<b>Distribution (customer and VTT)</b> SAFIR 2018 reference group 2, VTT archives				
<i>The use of the name of VTT Technical Research Centre of Finland Ltd in advertising or publishing of a part of this report is only permissible with written authorisation from VTT Technical Research Centre of Finland Ltd.</i>				

## Contents

---

Contents.....	2
1. Introduction.....	3
2. Existing software applicable to level 3 PSA analyses.....	4
2.1 Level 3 PSA codes .....	4
2.2 Consequence assessment codes relevant to level 3 PSA analyses.....	5
2.3 Atmospheric dispersion codes .....	5
2.4 Evacuation analysis programs .....	6
2.5 Geographic information systems.....	6
2.6 Database management systems.....	7
2.7 Mathematical and statistical software.....	7
3. Architectural views .....	7
3.1 Functional view .....	8
3.1.1 Mapping of level 3 PSA requirements to functionalities .....	10
3.2 Information view.....	12
3.3 Other viewpoints .....	14
4. Interfaces.....	14
4.1 Interface between levels 2 and 3 .....	14
4.2 Interfaces of components.....	14
4.3 Data interfaces.....	16
5. Conclusions .....	16
References.....	17

## 1. Introduction

---

The FinPSA software suite for probabilistic safety analysis (PSA) presently consists of FinPSA level 1 and FinPSA level 2. However, PSA has also a third level, where the consequences of a radioactive release from a nuclear power plant are assessed. Several level 3 PSA codes exist, but even the newest of them stem from the 1990's. Two codes for level 3 analyses have been developed at VTT. ARANO stems from the 1970's and is outdated in some respects (e.g. wind direction cannot be changed); Valma is from the 1990's but has not been developed primarily for level 3 PSA analyses. Therefore it is natural to consider constructing a new level 3 PSA code.

Such a new code would have several advantages:

- It could take latest developments in atmospheric modelling, dose estimation, and other central level 3 fields into account
- It could be built to utilize increased computational and data storage capacities of modern computer systems
- It could integrate seamlessly with FinPSA levels 1 and 2, thus complementing them to form a part of a PSA code suite (to the present author's knowledge, no PSA code suite exists that would cover levels 1-3 in an integrated manner)

This report provides a first sketch of an architecture for such a code. The purpose of the sketch is to provide initial information and view on which discussions with code stakeholders – foremostly Finnish NPP companies and regulatory agency STUK, but also their Nordic counterparts, international organizations such as OECD and IAEA, potential developers of the code, etc. – could be based on.

## 2. Existing software applicable to level 3 PSA analyses

---

It is evident that a new level 3 code should utilize existing applicable software when it is sensible and possible. We take a brief look at available software that could be used in level 3 PSA analyses, or at least parts of the analyses. This review is limited to code directly applicable for level 3 PSA purposes, relevant free and open source software (FOSS), and software that VTT administers. In addition to the software listed below, FinPSA levels 1 and 2 are available for fault and event tree modelling and analysis.

Selection of software to be included in FinPSA level 3 should take into account various factors. Price and license conditions are naturally central; inclusion of external software should not cause any extra payments or obligations to FinPSA level 3 license. From this point, FOSS is ideal. Suitability to level 3 PSA analyses is another; in this respect, quality of results and computational burden are essential. Suitability as a part of FinPSA Level 3 is also essential. All but the simplest software have a presumed architectural pattern that may be difficult to violate (Bass et al. 2003); it may be necessary to either adapt FinPSA level 3 to that pattern or isolate the software from the rest of FinPSA Level 3 in some fashion. Software that causes only minor (if any) architectural commitments is preferable.

### 2.1 Level 3 PSA codes

Several codes exist for more or less general level 3 PSA analyses.

- COSYMA (Jones et al. 1996), and its simplified PC version PC-COSYMA, is a code developed during the 1980s and 1990s for probabilistic off-site consequence analysis. It consists of several programs that allow the analysis of atmospheric dispersion, and early and late health effects taking the most common countermeasures into account. It is currently used by countries such as South Africa and the Netherlands to perform required Level 3 PSA analyses. The program is however, no longer supported.
- ARANO is a Finnish program which uses a Gaussian plume model for calculating population doses and health effects. It was originally developed at VTT in the 1970's. It is very fast and still used, although it has certain limitations, e.g. not being able to change wind direction, and not being able to take area topography into account.
- Lena is a simple standalone code developed in Sweden in the late 1980's. It is meant for quickly assessing ongoing or postulated nuclear accidents by emergency preparedness organizations,
- MAAP was developed in the late 1980's for the US Nuclear Regulatory Council. As all NRC-developed computer codes, it is available to participating countries in the Cooperative Severe Accident Research Program (CSARP), and is distributed much like the severe accident analysis program MELCOR. A fee applies.
- OSCAAR is a probabilistic consequence code developed in Japan in the 1980's. It is still in use.

Unfortunately, none of these codes seems to be free and open source.

## 2.2 Consequence assessment codes relevant to level 3 PSA analyses

RODOS is a decision support system for accident management developed in the 1990's and 2000's. The decisions concern various kinds of early and late countermeasures, such as evacuation, sheltering, distribution of iodine tablets, food bans, selection of crops etc. It is primarily meant for real-time decision-making; however, a statistical analysis tool for countermeasure planning has been implemented recently, making it more suitable for probabilistic analyses.

JRODOS is a version of RODOS written in Java. The main part of JRODOS was completed in 2009, but the code is continuously being updated. JRODOS is free and open source, and available upon request.

RODOS/JRODOS is not suitable for level 3 PSA in itself, because it does not handle probabilistic input or analyses. However, JRODOS contains source code for many level 3 relevant tasks for which it is otherwise hard to find software implementations, or even applicable theory. Such code includes population dose estimation, the handling of late countermeasures (land decontamination, selection of crops etc.), and aquatic dispersion.

VALMA is an emergency preparedness tool developed at VTT that provides a user interface to the atmospheric dispersion code SILAM (see section 2.3 below), with population dose estimation capabilities. It is deterministic, but can be programmed to conduct Monte Carlo simulations. VALMA is neither open-source nor free.

## 2.3 Atmospheric dispersion codes

Tens of atmospheric dispersion codes exist; for listings, see [http://en.wikipedia.org/wiki/List\\_of\\_atmospheric\\_dispersion\\_models](http://en.wikipedia.org/wiki/List_of_atmospheric_dispersion_models) or <http://pandora.meng.auth.gr/mds/strquery.php?wholedb>. However, most of these codes have a limited applicability to level 3 PSA due to constraints on the kinds of releases considered, and limitations on the range of reliable dispersion estimates (typically only up to 20-30 kilometres from release source). Some more notable atmospheric dispersion codes:

- SILAM is an atmospheric dispersion code developed by the Finnish Meteorological Institute (FMI). It is free and open source (available by request to FMI). It can analyse releases from a variety of sources, handles dispersion on a wide range (from 1 km to global), takes atmospheric chemistry and physics (e.g. decay of radioisotopes) into account, and can assimilate weather measurements into the dispersion estimates. Its main drawback in level 3 analyses is the computational burden of extensive analyses.
- AERMOD is a widely used atmospheric dispersion code. It can handle point sources such as typically release sources in nuclear accidents. It uses Gaussian dispersion for stable (i.e., low turbulence) atmospheric conditions and non-Gaussian dispersion for unstable (high turbulence) conditions. It can also handle wet and dry deposition of aerosols.
- CALPUFF is a non-steady-state puff dispersion model that simulates the effects of time- and space-varying meteorological conditions on pollution transport, transformation, and removal. It can be applied for long-range transport and for complex terrain.

From the Finnish point of view, SILAM is the most promising of these. It is accurate and reliable. However, the computational burden it causes leads to the idea that the new level 3 code could make use of two atmospheric dispersion codes: SILAM for computations requiring more accuracy, and for far fall-out (more than 50 kilometres) assessments, and

some other code for situations where approximate atmospheric dispersion estimates will do, and where the analysis spans only about 50 kilometres or less from the release source.

## 2.4 Evacuation analysis programs

Evacuation is arguably the most significant countermeasure to be analysed in level 3 PSA. There are some codes designed specifically for analysis of regional evacuation. I-DYNEV is a public domain program that has been available through the Federal Emergency Management Agency (FEMA) (Urbanik 2000). However, search on the FEMA www site produced only one reference to the program, a publication from the year 1984; thus it remains unclear whether the program is still available. OREMS (Oak Ridge Evacuation Modeling System) is available at <http://emc.ornl.gov/products/oak-ridge-evaluation-modeling-system.html>, but it is not free or open source. EMBLEM2 is a regional evacuation analysis program created for analysing hurricane evacuation (Lindell 2008). It uses a relatively simple model where evacuation time estimates are calculated by using four evacuation route system parameters, 16 behavioural parameters, and five evacuation scope/timing parameters. The applicability of EMBLEM2 to analysis of evacuation in nuclear emergencies is yet unclear.

One possibility would be to use a traffic simulation code by modifying it to account for the specifics of evacuation traffic. SUMO (Simulation of Urban Mobility) is an open source, highly portable, microscopic road traffic simulation package designed to handle large road networks (Behrisch et al 2011).

## 2.5 Geographic information systems

A geographic information system (GIS) is an information system designed to capture, store, manipulate, analyse, manage, and present all types of spatial or geographic data. In level 3 PSA analyses, a GIS can be used in many ways, such as land use and population data management, and economic and agricultural consequence assessment (e.g. estimate the amount of fields contaminated above some threshold value, given the spatial distribution of ground deposition).

Wikipedia

([https://en.wikipedia.org/wiki/List\\_of\\_geographic\\_information\\_systems\\_software#Open\\_source\\_software](https://en.wikipedia.org/wiki/List_of_geographic_information_systems_software#Open_source_software)) lists 13 free and open source (FOSS) GIS tools, and in addition many other useful tools for geospatial information processing. Two FOSS GIS programs, QGIS and GRASS, stand out as candidate providers of GIS services for FinPSA level 3. Further analysis is needed on which of these should be the main GIS program used by FinPSA level 3. Fortunately, the two programs can work together, and interfaces have been developed that enable the use of one from the other. In addition, there are other useful FOSS codes for geospatial information processing, such as GDAL (<http://www.gdal.org/>), a translator library for raster and vector geospatial data formats.

QGIS (<http://www.qgis.org/en/site/>) is a free and open-source geographic information system. It is potentially very useful in a number of tasks related to level 3 PSA:

- Composition and visualization of maps, e.g. contaminated areas
- Representation of physical geography (topography, vegetation, water systems etc.) needed in calculating atmospheric and aquatic dispersion, and human geography (population densities per area, land use, building data, road networks, economic activity etc.) needed in countermeasure computations and in consequence estimation (population doses, lost production etc.).
- Calculation of spatial statistics, e.g. the total area of agricultural land contaminated above a given threshold, counting the number of people living in a certain contaminated area etc.



- It uses Python as its scripting language. As Python is the language used also in SILAM, and may be considered as the implementation language of FinPSA level 3, this provides significant synergy advantages.

GRASS (<https://grass.osgeo.org/>) is, like QGIS, a free and open-source geographic information system. In addition to map creation and manipulation capabilities, it contains e.g. map-related image processing functionality. It also has an interface to the FOSS statistical package R.

## 2.6 Database management systems

Large amounts of data are associated with level 3 PSA, both on the input (population and geographic data, weather data) and output (atmospheric dispersion results, dose assessment results). A proper database management system (DBMS) would facilitate the management of such data. Some prominent FOSS DBMS systems are MySQL, MariaDB, SQLite and PostgreSQL. Of these, SQLite and PostgreSQL stand out because FOSS spatial extensions (Spatialite and PostGIS, respectively) have been developed for them.

## 2.7 Mathematical and statistical software

No FOSS software that would be directly targeted to mathematical and statistical analyses on level 3 PSA has been identified. Instead, general-purpose mathematical and statistical codes exist. R is at present perhaps the most popular program for statistical analyses and graphics. It has good capabilities for spatial statistics both directly and through packages. NumPy is a Python programming language extension that brings to it multidimensional arrays and matrices, along with a large library of high-level mathematical functions. SciPy is a Python library for scientific and technical computing. It has modules for optimization, linear algebra, integration, interpolation, special functions, fast Fourier transform, signal and image processing, ordinary differential equation solvers, among others. Other relevant Python libraries are matplotlib (plotting of mathematical and statistical graphics), pandas (data manipulation and analysis), and SymPy (symbolic computation). Gnu Octave and Scilab are numerical codes that can be considered to be FOSS alternatives to Matlab. Also machine learning software such as Weka and RapidMiner, and data mining software such as KNIME and OpenNN, could find applications in the analysis of data produced by level 3 PSA analyses.

## 3. Architectural views

---

We adapt the viewpoints (and views) approach to architectural description, introduced by (Kruchten 1995). We use the viewpoints specified in (Rozanski and Woods 2005). A viewpoint is a representation of one or more structural aspects of an architecture. A viewpoint is a collection of patterns, templates and conventions for constructing one type of view.

The viewpoints defined by Rozanski and Woods are functional, information, concurrency, development, deployment and operational views.

Before going to individual viewpoints, an overview of the main components of the architecture and the data flows between them is presented in Figure 1. The code receives source term from level 2 PSA. This is used in calculating atmospheric and aquatic dispersion estimates, which are used in assessing population doses and environmental and agricultural consequences. Population dose calculation takes also the effect of early and late countermeasures into account.



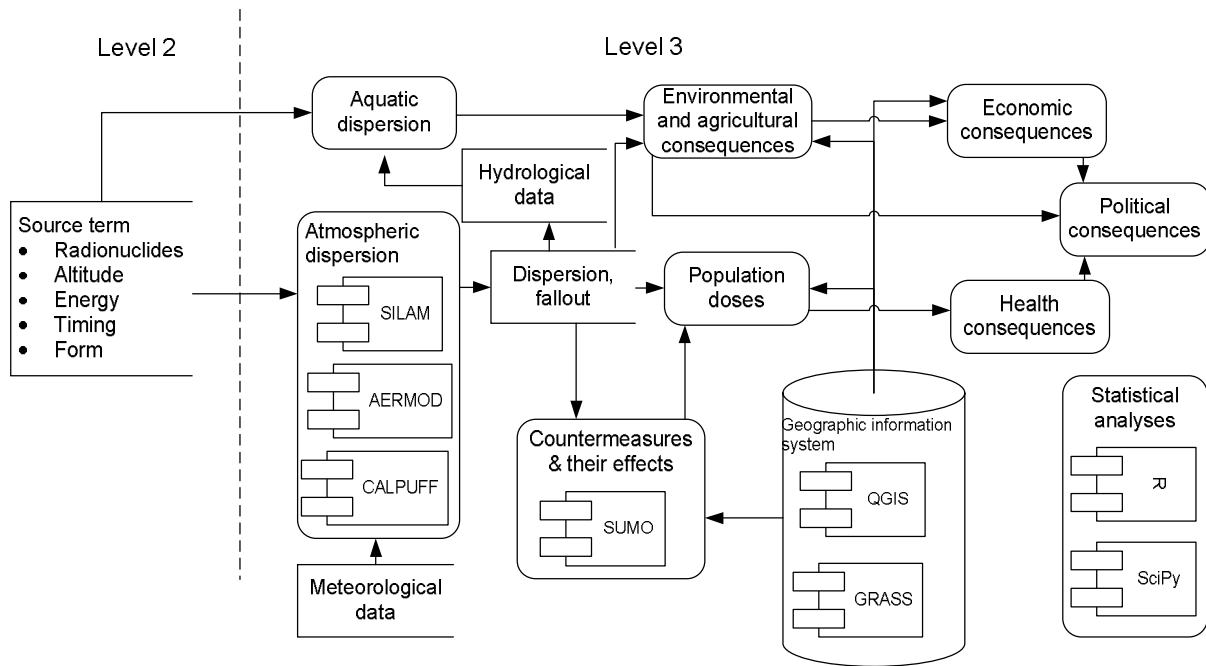


Figure 1. An overview of the level 3 code architecture

### 3.1 Functional view

The purpose of the functional view is to describe the system's functional elements, their responsibilities, interfaces, and primary interactions. The main functional view of the code is represented in Figure 2.

In terms of architectural style, the architecture resembles the tiered approach (see, e.g., Rozanski and Woods 2005). It has a presentation tier (user interface), a process tier (the control component that integrates the actions of individual components to a whole), a computation tier (the individual components/programs that handle the computations such as atmospheric dispersion, dose estimation etc.), a data access tier (interfaces to the databases/files/etc. that provide and store data for the code), and a data storage tier (the databases/files/etc. in which the data needed and produced resides).

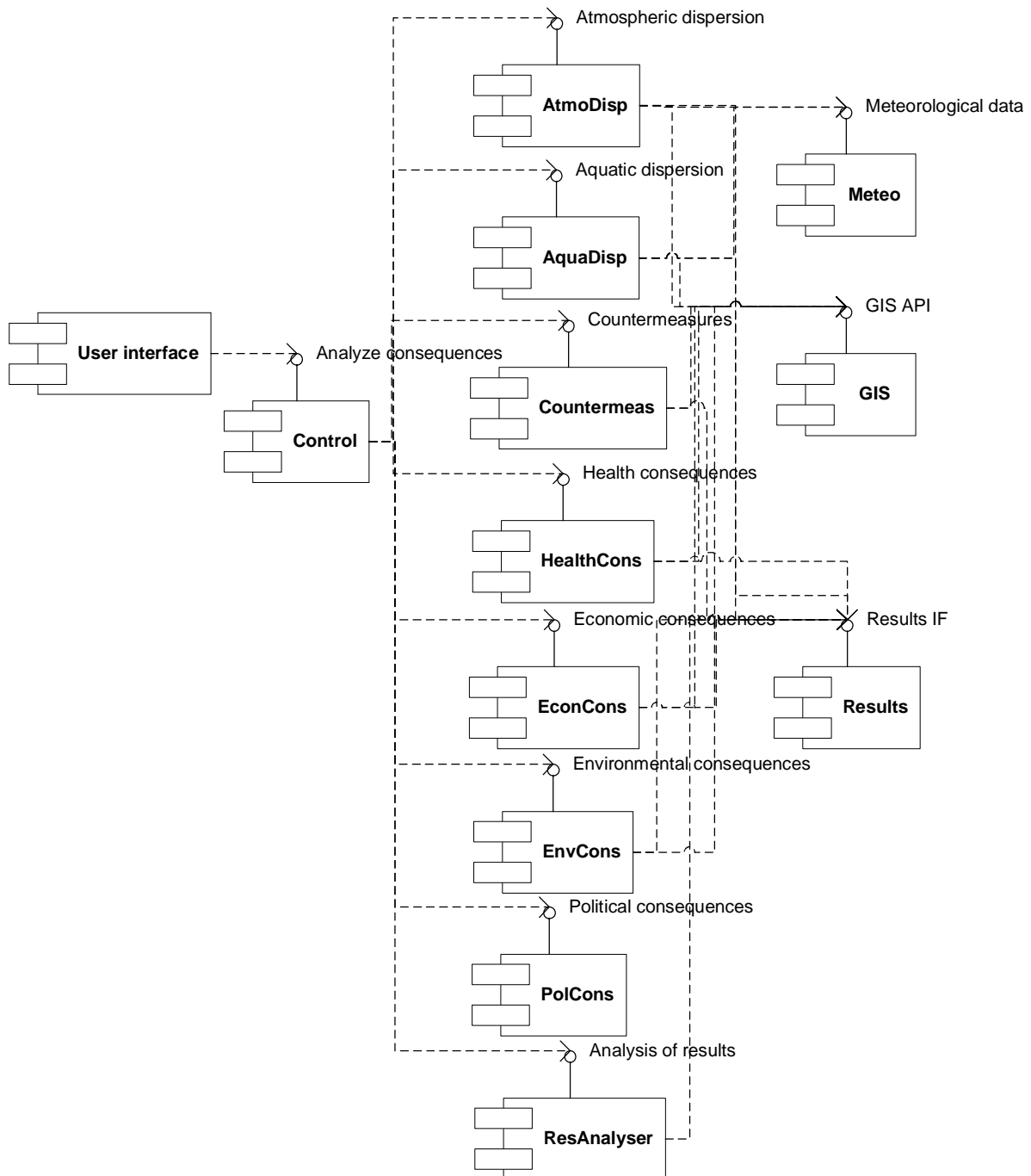


Figure 2. The main components and functional dependences of the code

The main components of the system are as follows:

- User interface takes care of the interaction with the user: taking user's commands and displaying the results. In addition to a graphical user interface, a line command interface may be considered for batch job execution.

- Control combines the different computations into a level 3 PSA analysis process. It takes care that the different level 3 tasks (atmospheric dispersion computation, dose assessment etc.) are executed in the right order, and that all tasks have all the data that they need available. It also takes care that the analysis results are analysed in the way that the user wants.
- AtmoDisp handles the atmospheric dispersion computation. Since the computation is actually performed by some external code (e.g. SILAM, CalPuff, Aermoc), it transforms data to the form used by the external code, calls the code with suitable parameters, and transforms the results to a form that can be utilized by other components. The external code used for atmospheric dispersion computation may be chosen by the user at runtime.
- AquaDisp handles the aquatic dispersion computation. Since no proper aquatic dispersion code seems to be available free and open source, the actual computation may have to be implemented as a part of FinPSA level 3.
- Countermeas computes the effect of countermeasures selected by the user, using atmospheric and aquatic dispersion results, population data etc. as inputs. Evacuation computations may be handled by an external program, but the computation of all other countermeasures has to be implemented in FinPSA level 3.
- HealthCons assesses population and individual doses through different pathways, and based on this, estimates early and late health consequences.
- EconCons assesses the economic consequences of a release, such as depreciation in property value, lost production, value of contaminated land, price of decontamination efforts etc.
- EnvCons assesses environmental and agricultural consequences, such as the amount of field/forest area contaminated.
- PolCons supports the analysis of political and societal consequences.
- ResAnalyser analyses the results produced by other components. This analysis includes making summaries, statistical analyses, uncertainty assessments, visualization, and meta-analyses (analysis of results from several scenarios or several NPP sites).
- Meteo is the manager of meteorological and hydrological information used in the analyses. This information may reside in a database, in files, or be available through a network. It serves the analysis needs of AtmoDisp and AquaDisp.
- GIS is the geographic information system component of FinPSA level 3. It acts as a front end to an open source GIS such as QGIS or GRASS or both. It may utilize databases to store and retrieve geographical data. It manages population, topographical, vegetation, land use, and road network data.
- Results manages the repository of analysis results. Each component of the analysis tier submits its results to the results component, which stores them and returns them for analysis, visualization etc. in the ResAnalyser component.

### 3.1.1 Mapping of level 3 PSA requirements to functionalities

In (Karanta 2014), requirements for a level 3 PSA code were outlined. In this section, the functional requirements of that document are mapped to the components listed above.

Table 1. Functional requirements of (Karanta 2014) and respective components

Requirement	Implementing component(s)
3.1.1 Atmospheric dispersion and ground deposition as a function of time, both for short and long range.	AtmoDisp, using external atmospheric dispersion codes such as SILAM (for long-range and high-accuracy), and AERMOD and CALPUFF (for short-range and approximate)
3.1.2 Aquatic dispersion and deposit as a function of time at different parts of the water system(s)	AquaDisp, using a suitable external aquatic dispersion code if such a code is found.
3.1.3 Radiation from stationary sources, possibly combined with atmospheric and aquatic dispersion	HealthCons for radiation estimation, AtmoDisp for atmospheric dispersion, AquaDisp for aquatic dispersion
3.1.4 Early and late health effects assessment, including health-related risk measures	Results of atmospheric and aquatic dispersion computations as inputs. Countermeasures for taking the effects of countermeasures into account. HealthCons for population dose estimation and early and late health effects assessment.
3.1.5 Agricultural and environmental effects assessment	Results of atmospheric and aquatic dispersion as inputs. EnvCons for consequence assessment.
3.1.6 Economic effects assessment	Results of atmospheric and aquatic dispersion, countermeasures, health and environmental consequences assessment, and political consequences assessment as inputs. EconCons for cost estimation and the assessment of other economic consequences.
3.1.7 Estimation and evaluation of political consequences	Results of atmospheric and aquatic dispersion, countermeasures, health and environmental consequences assessment as inputs. PolCons for political consequences assessment.
3.1.8 Importance measures estimation	ResAnalyser for estimation. May need Control to coordinate computations in other components (AtmoDisp, HealthCons etc.).
3.1.9 Uncertainty and sensitivity analysis	ResAnalysis for estimation. Control for managing computations in other components (AtmoDisp, HealthCons etc.).
3.1.10 Probabilistic analysis of consequences	Control (using AtmoDisp, AquaDisp etc.) for Monte Carlo simulation. If more sophisticated probabilistic modelling is needed, for example FinPSA level 2 might come to question.
3.1.11 Deterministic analysis of consequences	Control, using AtmoDisp, AquaDisp etc.
3.1.12 Integrated probabilistic risk analysis on PSA levels 1-3	Software that is built on top of FinPSA levels 1, 2 and 3, and uses all of these. On FinPSA Level 3 part, that software will interface with Control.
3.1.13 Reporting of analysis results	ResAnalyser, using results of other components.

### 3.2 Information view

The information view describes the way that the architecture stores, manipulates, manages, and distributes information. In this section, only the most data-intensive parts of the system are described. These include weather, geographic and population data on the input side, and atmospheric and aquatic dispersion, and dose assessment results on the output side.

Concerning weather data, it remains to be decided whether it should remain in a database management system, in text files, or both. Using a spatial database and GIS would allow analysis of weather data for, e.g., finding spatial patterns in weather when data from several weather masts in a certain region are available. However, the weather information depicted in Figure 3 needs to be present:

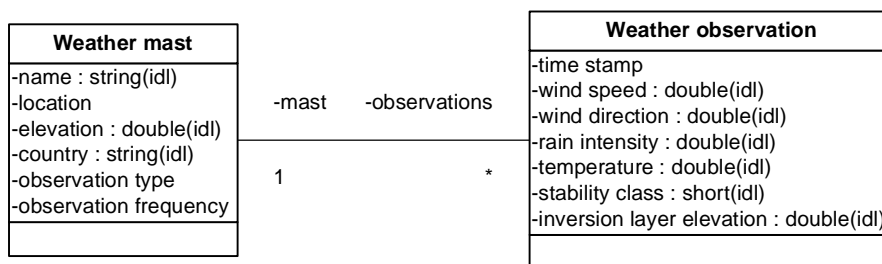


Figure 3. Weather data model for level 3 PSA purposes

Weather mast is a facility for weather variable measurement. Location gives the coordinates of the weather mast; elevation the observation elevation from ground level; observation type can be e.g. “average”, “instantaneous”, “maximal”, “minimal”; and observation frequency can be e.g. “hourly”, “daily”. Time stamp would provide the time of the observation.

A design issue needs to be specified: is there a need for weather data that is not associated with a single weather mast, e.g. data that consists for example from statistical averages over a certain region.

Weather forecasts, or weather scenarios, may be used in level 3 PSA analyses, too. These, too, can be expressed in terms of weather mast(s) and associated weather observation data. To distinguish these from actual observations it may be necessary to e.g. create imaginary weather masts and associate forecasts and scenario data with these, and to label these imaginary masts as virtual.

A general view of geographic data needed in level 3 analyses is given in Figure 4. It consists of items related to physical geography (topography, water systems), human geography (population data, road networks, land use), and biogeography (vegetation). More detailed modelling will add detail, such as modelling water systems and road networks as directed graphs, specifying land use attributes such as property prices for real estate in residential and industrial areas, production net worth of companies in an industrial area etc.

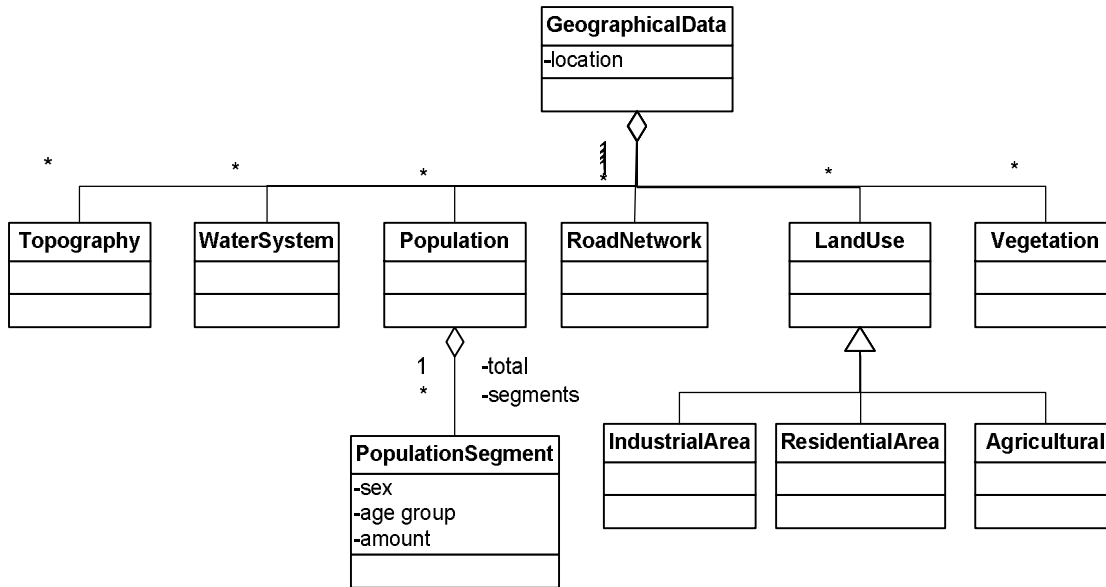


Figure 4. Geographic data model for level 3 PSA purposes

Of the processes in FinPSA Level 3, the most data-intensive is the estimation of population doses from atmospheric dispersion estimates and population data. This is depicted in the data flow diagram Figure 5.

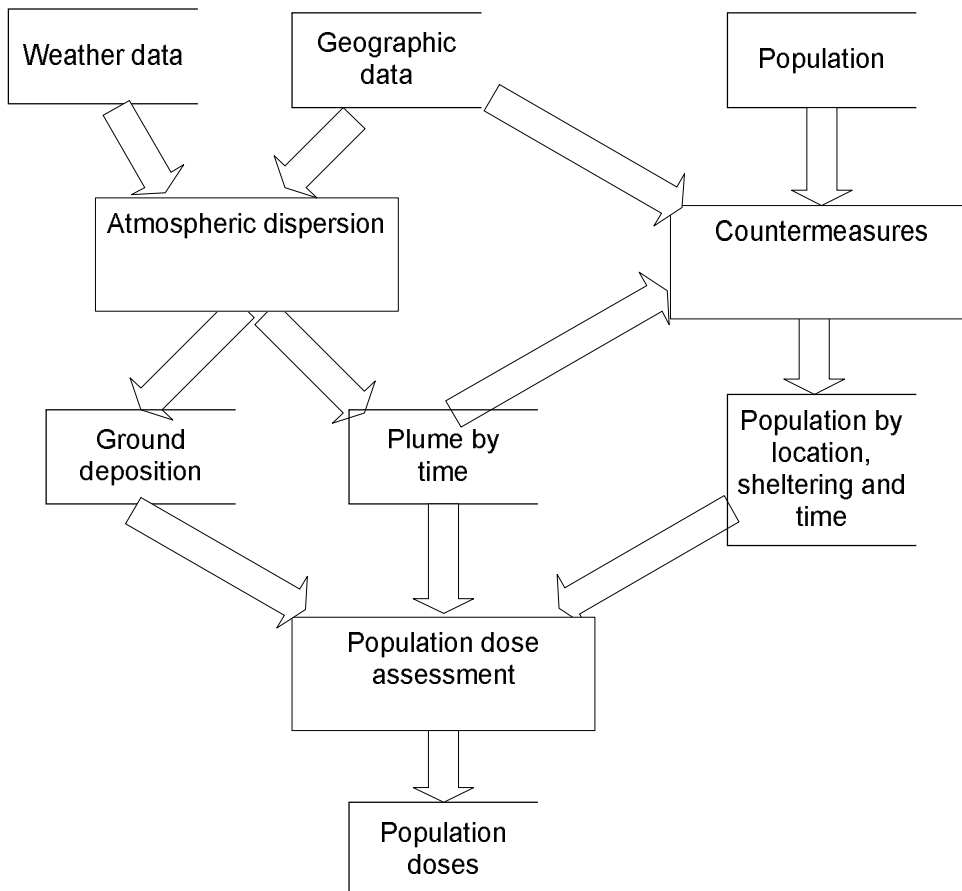


Figure 5. Data flow of population dose estimation

The basic input to population dose estimation process consists of weather data; geographic data (e.g. elevation and vegetation data) may be used also if the atmospheric dispersion

model can make use of it. The atmospheric dispersion assessment produces plume and ground deposition estimates by time and location. On the other hand, the effects of countermeasures are estimated from population and geographic data; the progression of the plume may affect the timing of countermeasures, and may therefore also act as an input to countermeasures calculation. From estimates of plume, ground deposition, population location and population sheltering by time, population dose assessment can be made. This assessment may be conducted by population segment, by organ etc. Its result is a set of population dose estimates.

### 3.3 Other viewpoints

In addition to the functional and information viewpoints presented above, Rozanski and Woods specify other views that are not germane at this stage of system design. They will be briefly handled here, and receive more detailed treatment once a decision is made to develop FinPSA level 3.

Concerning the concurrency viewpoint, FinPSA level 3 will be designed primarily to be used on a single workstation, but with the ability to perform heavy computations in a parallel manner in several computers connected to a network. Since SILAM will most probably be an atmospheric dispersion code used by FinPSA Level 3, atmospheric dispersion computations will likely be the computationally most demanding ones performed by FinPSA level 3, and SILAM has been successfully parallelized, its parallelization model may be used in FinPSA level 3, too.

Issues related to the development and operational viewpoints will be decided later.

Concerning the deployment viewpoint, it is likely that FinPSA level 3 will run on the Windows platform, as do FinPSA levels 1 and 2. The code will most likely use several of the FOSS programs introduced in section 2.

## 4. Interfaces

---

This chapter contains sketches of the most important interfaces within FinPSA level 3. The purpose of presenting these interfaces is to illustrate the functionality of its components and facilitate communication with code stakeholders rather than act as a rigorous interface specification describing interface identity, resources provided, and other interface elements as detailed by (Clements et al. 2003). Rigorous interface specifications will be constructed if it is decided that FinPSA Level 3 will be built.

It is expected that the interfaces in FinPSA level 3 will be implemented as ordinary subroutine/function calls. Nevertheless, this is a design issue that can be decided later.

### 4.1 Interface between levels 2 and 3

The interface between PSA levels 2 and 3 has been defined in (Karanta 2013). The input from level 2 to level 3 consists of the source term (amounts of each radionuclide released, start time, release height from ground level, release temperature), and possibly also a particle size distribution of the source term.

### 4.2 Interfaces of components

The main services provided by each component are listed in Table 2.



Table 2. Major interfaces of FinPSA Level 3, on the main functionality granularity.

Component	Interface	Description
AtmoDisp	atmospheric dispersion	Subroutine assess_atmospheric_dispersion(source_term,weather,geography) estimates atmospheric dispersion and ground deposition as functions of time
AquaDisp	aquatic dispersion	Subroutine assess_aquatic_dispersion(aquaticSourceTerm,waterSystemGeography) estimates aquatic dispersion and deposit in the water system as a function of time
Countermeas	countermeasures	Subroutine assess_evacuation(atmosphericDispersion,populationData,geographicData) estimates the amount of population by location as a function of time. Subroutine assess_sheltering(atmosphericDispersion,populationData,geographicData) estimates population by location and time, and their protection against radionuclides by each pathway as a function of time Estimation of the effects of other countermeasures similarly.
HealthCons	health consequences	Subroutine estimate_population_dose(atmosphericDispersion,populationData, optional countermeasuresResults,pathways) estimates dose by population segment, organ and each pathway considered. estimate_health_consequences(atmosphericDispersion,populationData, optional countermeasuresResults,pathways,timespan) estimates population dose and assesses health effects (radiation sickness, cancers) for the given time span.
EconCons	economic consequences	Subroutine estimate_economic_consequences(healthEffects,environmentalEffects,countermeasures, optional politicalConsequences) estimates the costs induced by each consequence.
EnvCons	environmental consequences	Subroutine estimate_environmental_consequences(atmosphericDispersion,populationData,geographicData,consequenceClasses) estimates the environmental consequences defined by consequenceClasses. These could be, e.g., the area of agricultural land where the activity of ground deposition per square meter exceed a given threshold, migration of radioactive substances in the biosphere etc.
PolCons	political consequences	assess_political_consequences(healthCons

		<p>sequencesResults,environmentalConsequencesResults,countermeasuresResults) will provide an assessment tool where the user(s) can estimate the probabilities of various political consequences (e.g. prohibition of operation of some nuclear power plants) in a structured and systematic manner.</p>
ResAnalyser	analysis of results	<p>computeImportanceMeasures(analysisResults,measuresNeeded) estimates the importance measures that the user requests. report_analysis_results(analysisResults,resultsRequested) filters and summarizes analysis results, and returns a textual report or document listing the requested results.</p>
Meteo	meteorological data	<p>get_meteorological_data(area,timespan) returns meteorological data for the given area and timespan set_meteorological_data(newMeteoData) updates the meteorological database by the new meteorological data.</p>
GIS	GIS API	<p>get_GIS_data(dataKind,range) returns geographical or population data for the given range. set_GIS_data(newData) updates the geographic information system database with new data.</p>
Results	results IF	<p>set_results(newResults) updates the result database with new results from analyses get_results(resultKind,range) returns the requested results from the results database.</p>

### 4.3 Data interfaces

It is probable that the large amounts of data associated with weather observations, geographic and population information, and atmospheric dispersion and dose assessment results will necessitate a database. It might turn out that the volume of data makes it reasonable to have two, instead of one, databases so that the databases can be installed on separate servers. This may solve some performance and memory capacity problems, and may also facilitate the conduct of level 3 analyses by a workgroup. A suitable arrangement might be that one database contains input data and another results data.

In FinPSA Level 3, these databases would be queried and manipulated by the Meteo, GIS and Results components.

Smaller datasets, such as those associated with radionuclide properties or economic cost estimates, may also be stored in ordinary files.

## 5. Conclusions

---

A draft architecture for a level 3 PSA code that may be developed in the future has been sketched. Applicable software has been reviewed, the most important viewpoints, functional

and information, have been outlined, level 3 PSA code requirements have been mapped to functionalities, and an initial attempt has been made to specify interfaces for different components of the architecture.

It seems that a large part of the functionality needed in a level 3 PSA code is already available in free and open-source programs. This applies especially regarding atmospheric dispersion, geographic information management, database management systems, and mathematical and statistical analysis. Also some functionality directly relevant to level 3 is available. For such functionality, FinPSA Level 3 will act as a software glue that integrates these codes into a whole and coordinates their action. The most important functionality that is not available as FOSS is related to more sophisticated methods of dose assessment; much of this functionality has to be implemented as a part of the construction of the construction of FinPSA Level 3.

The proposed architecture is based on a tiered structure. Advantages of this kind of architectural style include a clear separation of concerns between tiers, and potential of reusability of the simpler tiers. The latter would come into use if e.g. a lightweight level 3 PSA code, consisting of atmospheric dispersion, dose assessment and health consequences estimation would be needed; then, the same components could be used in both this code and a more comprehensive level 3 PSA code.

Construction of a new level 3 PSA code would bring many advantages. However, at the time of writing, it is unclear whether such a new code is needed. Finding this out requires communication with code stakeholders: nuclear power plant companies and regulatory agencies. One of the purposes of this document is to facilitate that communication.

If a decision is made to develop a new level 3 PSA code, a more detailed architecture specification is needed. The specifications presented in this document will in that case most likely be changed, but then this document will have served its purpose of facilitating communication and acting as a basis of further development.

## References

---

- Len Bass, Paul Clements, Rick Kazman. Software architecture in practice, 2<sup>nd</sup> edition. Addison-Wesley 2003.
- Michael Behrisch, Laura Bieker, Jakob Erdmann, Daniel Krajzewicz. SUMO – Simulation of Urban Mobility, an overview. SIMUL 2011, The Third International Conference on Advances in System Simulation, pp. 63-68.
- Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Robert Nord, Judith Stafford. Documenting software architectures – views and beyond. Addison Wesley 2003.
- J. A. Jones, I. Hasemann, J. van der Steen. COSYMA, a mainframe and PC program package for assessing the consequences of hypothetical accidents. Proceedings of the 3<sup>rd</sup> Regional meeting of the Nuclear Society of Slovenia (Nuclear Energy in Central Europe), Portoroz, Slovenia, 16-19 September 1996, 9 pages.
- Ilkka Karanta. Level 3 PSA from a software architecture point of view. VTT Research Report VTT-R-01071-13, VTT, Espoo 2013, 13 pages.
- Ilkka Karanta. Software requirement specification, project PRADA. Version 0.2, 5.5.2014, 22 pages.

Philippe Kruchten. Architectural Blueprints—The “4+1” View Model of Software Architecture. IEEE Software Vol. 12 No. 6 (1995), 42-50.

Michael K. Lindell. EMBLEM2: An empirically based large scale evacuation time estimate model. Transportation Research Part A 42 (2008) 140–154.

Nick Rozanski, Eoin Woods. Software systems architecture – working with shareholders using viewpoints and perspectives. Addison-Wesley, Upper Saddle River 2005.

Thomas Urbanik. Evacuation time estimates for nuclear power plants. Journal of Hazardous Materials, Vol. 75 (2000), 165-180.