Project	SEAMLESS No 101004032	**Start / Duration**	1 January 2021/ 36 Months
Dissemination	Public	**Nature**	**Other/software**
Date	2021-09-27	**Version**	**1.0**

Deliverable D2.1

Prototype core system

Deliverable Contributors:	Name	Organisation	Role / Title
Deliverable Leader	Karsten Bolding	BB	Task Leader
Contributing Author(s)	Jorn Bruggeman	BB	Task Contributor
Reviewer(s)	Lars Nerger	AWI	WP2 Lead
Final review and approval	Stefano Ciavatta	PML	Coordinator

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004032.

Document History:

Release	Date	Reason for Change	Status	Distribution
1.0		Initial document, reviewed	Released	Public

To cite this document

This document describes the current software status for the core infrastructure of the prototype. Please cite the software SEAMLESS Ensemble and Assimilation Tool (EAT) publicly available in the repository https://github.com/BoldingBruggeman/eat along with this report Deliverable D2.1 “Prototype core system”

TABLE OF CONTENTS

1. Scope

Deliverable D2.1 is concerned with task 2.2 – Data assimilation functionality. Here it is in particular the subtask 2.2a – coupling of PDAF and GOTM (see below) providing the core infrastructure including documentation. Other subtasks of task 2.2, namely 2.2b – Implementation of simultaneous physical-BGC assimilation and 2.2c – Implementation of support for parameter estimation and correction by DA will be part of Deliverable D2.4 later in the project.

The deliverable is a software product with public availability from the GitHub repository https://github.com/BoldingBruggeman/eat. Further developments will be made available via the repository and will include developments in response to SEAMLESS partners using the software in other work packages.

This report provides a synthetic description of the back ground and design ideas. Any developer/user focussed documentation is provided as part of the Git repository.

2. Introduction

The prototype core system is a software deliverable comprising? a coupled 1-dimensional physical-biogeochemical model with data-assimilation capabilities.

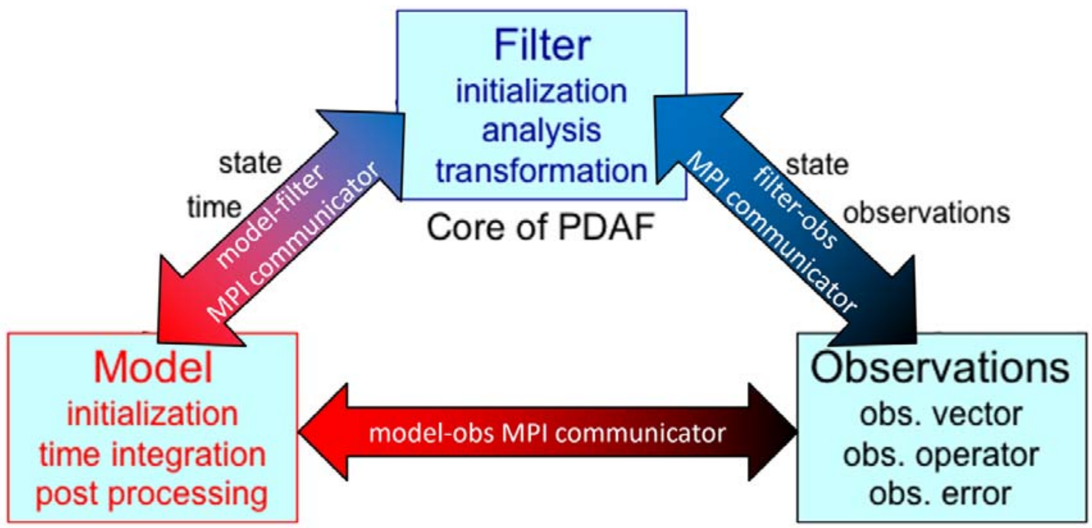
The software project is called SEAMLESS Ensemble and Assimilation Tool (EAT) and is public available on GitHub via – https://github.com/BoldingBruggeman/eat. EAT will be further developed during the course of the SEAMLESS project and the GitHub page will always have the authoritative version. EAT builds on other software projects – notably General Ocean Turbulence Model (GOTM – https://gotm.net), Framework for Aquatic Biogeochemical Models (FABM - https://fabm.net) and the Parallel Data Assimilation Framework (PDAF – http://pdaf.awi.de) and integrates these different components into a software product capable of doing data-assimilation simulations for any GOTM configuration where observations are available.

EAT is intended as both a development platform for testing new assimilation methods and new biogeochemical models – but – also as a production ready assimilation system for realistic 1D setups.

In its most basic form EAT is a protocol for information exchange between the three components constituting a data-assimilation system – 1) the model, 2) the assimilation method and 3) the observation handler. EAT is also an implementation of the protocol for GOTM/FABM/PDAF and a proof of concept reimplementation of the PDAF 2D tutorial case. EAT maintains the different components separate and all information exchange is done via a thin software layer utilising the Message Parsing Interface (MPI) standard to do the actual sending and receiving of data. As all information exchange is done via MPI – and the different components are not linked together to form a monolithic executable – each of the components can be implemented in any programming language supporting MPI. As illustration – the observation handler for GOTM is implemented both in Fortran and Python.

MPI uses the term – communicators – for grouping processes together where messages can be sent and received between the processes belonging to the group.

During EAT initialization 6 communicators are defined. These are communicators for observation processes (O), filter processes (F) and model processes (M) as well as for the combinations of observation-filter processes (OF), observation-model processes (OM) and finally model-filter processes (MF). The acronyms are used for logging purposes when running EAT.



The information exchange protocol is kept very simple and it might require updates in the future if use cases proves it necessary.

OM communicator: the observation handler sends observation times to all model processes.

OF communicator: the observation handler sends the number of observations (nobs) and observation vector to the filter process.

MF communicator: model processes send forecasts (in state vector form) to the filter process and after applying the filter process sends the analysis (also in state vector form) to each of the model processes.

The above is done every time observations are available. The observation handler signals no more observations by sending nobs=-1 to the filter process and a non-valid time string to the model processes.

One design aim of EAT is to make as few changes to the original software components as possible and instead write small software interface layers around the core functionality of the different components. One advantage of this approach is that the basic components can be further developed without impacting EAT – as long as the application programming interface (API) does not change. To further emphasize this design approach GOTM and FABM are included in EAT as Git submodules (i.e. no changes required). PDAF requires a few more steps as it does not support software configuration using CMake (an advantage for cross platform support). To overcome this problem EAT comes with CMake configuration files that must be copied from EAT to where PDAF is un-packed. The specific files to copy are shown on the GitHub page.

3. Procedures

Information on obtaining, configuring and building EAT is available via https://github.com/BoldingBruggeman/eat. EAT is developed for people with experience in software development and numerical modelling.