Large-scale EXecution for Industry & Society

Deliverable D6.2

Pilot Improvements: Solutions Adopted

Co-funded by the Horizon 2020 Framework Programme of the European UnionGrant Agreement Number 825532
Grant Agreement Number 825532
ICT-11-2018-2019 (IA - Innovation Action)

DELIVERABLE ID | TITLE	D6.2 | Pilot improvements: solutions adopted
RESPONSIBLE AUTHOR	Natalja Rakowsky (AWI)
WORKPACKAGE ID | TITLE	WP6 | Earthquake and Tsunami large scale pilot
WORKPACKAGE LEADER	CEA
DATE OF DELIVERY (CONTRACTUAL)	31/03/2020 (M15)
DATE OF DELIVERY (SUBMITTED)	01/04/2020 (M16)
VERSION | STATUS	V2.2 | Final
TYPE OF DELIVERABLE	R (Report)
DISSEMINATION LEVEL	PU (Public)
AUTHORS (PARTNER)	N. Rakowsky (AWI), Thierry Goubier (CEA), Andrea Ajmar (ITHACA), Danijel Scholremmer (GFZ)
INTERNAL REVIEW	Florin Apopei (TESEO), Tomáš Martinovič (IT4I)

Project Coordinator: Dr. Jan Martinovič – IT4Innovations, VSB – Technical University of Ostrava
E-mail: jan.martinovic@vsb.cz, **Phone:** +420 597 329 598, **Web:** https://lexis-project.eu

DOCUMENT VERSION

VERSION	MODIFICATION(S)	DATE	AUTHOR(S)
0.1	Initial template	01/06/2019	Thierry Goubier (CEA)
0.2	Template adapted	26/11/2019	Natalja Rakowsky (AWI)
0.3	Structure + DSL added	13/2/2020	Natalja Rakowsky (AWI), Thierry Goubier (CEA)
0.4	Major contributions by all partners	28/2/2020	Natalja Rakowsky (AWI), Thierry Goubier (CEA), Andrea Ajmar (ITHACA), Danijel Schorlemmer (GFZ)
1.0	Recovering after problems with OnlyOffice	01/03/2020	Natalja Rakowsky (AWI)
2.0	Reviewed	19/03/2020	Florin Ionut (TESEO), Tomáš Martinovič (IT4I), Thierry Goubier (CEA), Natalja Rakowsky (AWI)
2.1	Related projects content after EAB review	26/03/2020	Thierry Goubier (CEA)
2.2	Final check	30/03/2020	Kateřina Slaninová (IT4I)

GLOSSARY

ACRONYM	DESCRIPTION
AOI	Area Of Interest
BPMN	Business Process Model Notation: a graphical notation to describe and model business processes.
CEMS	Copernicus Emergency Management Service
EQ	Earthquake
FEP	First Estimate Product
GIS	Geographical Information System: a database and additional functions for data with a geographical meaning and coordinates.
MOC	Model of Computation
QML	The name of an actor in the MoC producing a QuakeML file
QUAKEML	Quake Markup Language: a flexible, extensible and modular XML representation of seismological data, e.g. epicenter, hypocenter, magnitude.
RISE	Project in Horizon 2020: “Real-time earthquake rIsk reduction for a reSilient Europe”
RM	Rapid Mapping
SDF	Synchronous Data Flow
SEM	Satellite-based Emergency Mapping: the process of using remote sensing data to produce maps for emergencies
TSUNAWI	Tsunami simulation code developed at AWI

TABLE OF PARTNERS

ACRONYM	PARTNER
Avio Aero	GE AVIO SRL
AWI	ALFRED WEGENER INSTITUT HELMHOLTZ ZENTRUM FUR POLAR UND MEERESFORSCHUNG
BLABS	BAYNCORE LABS LIMITED
Bull/Atos	BULL SAS
CEA	COMMISSARIAT A L ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES
CIMA	Centro Internazionale in Monitoraggio Ambientale - Fondazione CIMA
CYC	CYCLOPS LABS GMBH
ECMWF	EUROPEAN CENTRE FOR MEDIUM-RANGE WEATHER FORECASTS
GFZ	HELMHOLTZ ZENTRUM POTSDAM DEUTSCHESGEOFORSCHUNGSZENTRUM GFZ
IT4I	VYSOKA SKOLA BANSKA - TECHNICKA UNIVERZITA OSTRAVA / IT4Innovations National Supercomputing Centre
ITHACA	ASSOCIAZIONE ITHACA
LINKS	FONDAZIONE LINKS / ISTITUTO SUPERIORE MARIO BOELLA ISMB
LRZ	BAYERISCHE AKADEMIE DER WISSENSCHAFTEN / Leibniz Rechenzentrum der BAdW
NUM	NUMTECH
O24	OUTPOST 24 FRANCE
TESEO	TESEO SPA TECNOLOGIE E SISTEMI ELETTRONICI ED OTTICI

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

EXECUTIVE SUMMARY

This deliverable describes the analysis, evolution, and planned evolutions of the main software components of the earthquake and tsunami large scale pilot of the LEXIS project. This pilot is a use case mixing up Cloud, Big Data and high performance computing components chained together in a flow with real time constraints, and exercise the federation of resources and technologies at the convergence of cloud, big data and high performance computing that the LEXIS project is building.

Position of the deliverable in the whole project context

The LEXIS project relies on three pilots to validate and deploy its technology and infrastructure advances, giving to each pilot its own work package. The work package 6 (WP6) is dedicated to the earthquake and tsunami large scale pilot. After the first Task 6.1 of scenarios and baseline requirements definition, this work package is running Task 6.2, a development task where individual components of the pilot are improved to benefit from the technology of the project, and to prepare them for integration on the LEXIS platform, integration effectively undertaken in Task 6.3.

Description of the deliverable

This deliverable contains an updated description of the tsunami and earthquake large scale pilot workflow, in particular, the scheduling groups in the workflow. It also described the work done on the model of computation adaptation to the LEXIS platform, analysis, and evolution of the various components of the pilot. Among those, the OpenBuildingMap update process, the ShakeMap generation, the loss assessment computations (at both aggregate and detailed), the TsunAWI fast and precise simulations, and a methodology to evaluate the gains of the pilot on the satellite-based emergency mapping process. Two additional past events are considered as data source, to better quantify the benefits of the pilot.

1 INTRODUCTION

The LEXIS project is developing solutions for federated HPC and Cloud resources, enhanced by burst buffer technology. This is comprised of three pillars, that is infrastructure, orchestration and data management. The project relies on three pilots representing different domains and exercising the LEXIS platform.

The pilot objective is to demonstrate a time constrained workflow for disasters, combining earthquake loss assessment and tsunami simulations into a flow targeting on-time availability of first estimates and detailed estimates, so as to support emergency response decisions and enhance the production of emergency mapping services.

It combines four main application components: A ShakeMap [1] code from the OpenQuake library [2] to compute ground-motion distributions for earthquakes; TsunAWI [1], a code simulating the propagation and inundation of an earthquake-triggered tsunami; OpenBuildingMap [1], a database of geographical information focusing on the built environment and an associated building classification and loss assessment method allowing detailed damage assessments at the building scale after an earthquake and tsunami event, and a satellite-based emergency mapping process producing post-disaster map products. Those four applications are integrated in a workflow described by the PolyGraph [3] model of computation, and implemented over the infrastructure, orchestration and data management of the LEXIS platform.

The pilot being based on the notion of providing timely results in an emergency situation triggered by a disaster event, a strong effort is made to ensure that early estimates can be provided in a very short time by optimising each of the relevant component, and finding innovative implementation solutions (aggregation, suitable indexing schemes), as well as ensuring via the model of computation that the orchestration of the pilot can be organized so as to respect deadlines. Additional reference datasets are considered, in particular to allow for a better understanding of the gains offered by this pilot compared to previous implementations of the production of emergency mapping services, and consideration is taken onto the resources needed to make this pilot sustainable in the long term.

1.1 RELATED PROJECTS

Solutions for Tsunami simulations under emergency conditions already do exist, such as the INATEWS system in Indonesia, using pre-determined scenarios and fast, approximate algorithms such as Eazywave on low resolution data. Experimental solutions with full simulations do exist, such as the system setup at the University of Tohoku [4], with good results in terms of computation acceleration. Those systems differ with our pilot in two ways: first, they are based on an on-line vision of real-time, which means making the algorithm as fast as possible to decrease the computation time, hoping that it will be enough to provide timely information; in this pilot, we really do consider the fact that first we have a deadline to meet, and second that simulations may be fast enough. The second difference is the representation used by the tsunami simulations. TsunAWI uses an unstructured mesh that is relatively hard to accelerate, whereas the Tohoku system uses the TUNAMI model, a set of regular grids with different resolutions that is relatively easy to vectorize; both models provide comparable results with the same initial data and topography resolution [5].

In the HPC for Urgent Computing Workshop at Supercomputing 2019, that the project co-organized along with the VESTEC [6] FET-HPC European project, publications listed various approaches to use HPC and Cloud systems for that kind of computing, from on-demand simulations on the Cloud [7] to accelerating tsunami simulations [8]. Apart from work focusing on interaction with fast running simulations for decision support [9], publications were concerned about running faster simulations (reducing time to solutions), this allowing more simulations to be run to allow to cope with source uncertainty, source uncertainties being an issue in the early stages of an earthquake event.

The usual way to compute an earthquake loss assessment, is to have a coarse exposure model (with built-in vulnerability functions) on a district level, with low spatial resolution (county or state-level), dependent on the administrative boundaries of the affected country or countries. Those boundaries are also the source of the data for the exposure model (buildings et al.) [10], [11], but also makes a uniform assessment between countries problematic because of the differences of what a district is. Going down to a building level as we are is only envisioned in the research community, and we will be the first to so increase the spatial resolution of the data. This will resolve differences between districts definition among countries, while still being able to report aggregates on a district level if needed. In all cases, computing results is fast due to the coarse granularity of the exposure models, and we will maintain the short computation times as appropriate even as we increase the spatial resolution.

This pilot uses a similar methodology with those other projects at the individual component level, which is to innovate in the development and implementation of new and improved algorithms to have shorter time-to-results. Where we differ is, for tsunami simulation, the high efficiency (low number of computations) for a given precision, for earthquake loss assessment, the vastly increased spatial resolution, and, for the overall flow, an effective and explicit awareness of real-time deadlines in the workflow, and so, sub-workflows taking explicitly that into account.

2 EVOLUTION OF THE WORKFLOW

As an essential component of the pilot, the workflow has been confirmed and evolved slightly since the previous update, documented in Deliverable D6.1 [12]. The main changes are that hypothetical components have been approved, some new technical solutions have been developed, and the overall scheduling of the workflow tasks has been completely defined during a December F2F meeting in Prague, and is reflected in Figure 1.

2.1 HIGH-LEVEL OVERVIEW

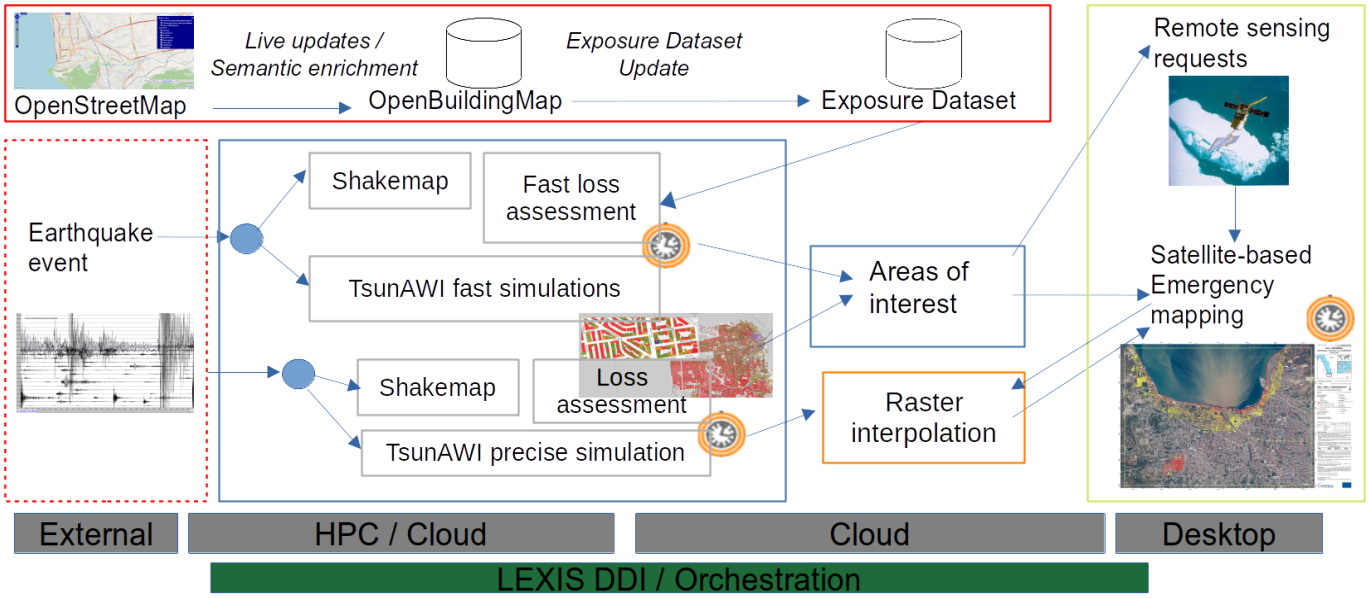

Figure 1 High-level view of the tsunami and earthquake pilot workflow.
The workflow is as currently set, with each box representing a different scheduling domain, and their approximate mapping onto the different type of resources of the LEXIS platform.

The high level overview follows the general description given in Deliverable D6.1, with some additional information and refinement. The core is still, upon an earthquake event, a ShakeMap generation and at least one fast TsunAWI tsunami simulation are started, followed by a fast loss assessment (aggregate information at the district level or similar entity), and this before the first deadline (triggering a tsunami warning or not). Then upon reception of the detailed earthquake moment tensor, an extensive TsunAWI simulation is run and a detailed loss calculation is started. All those tasks produce independent areas of interest that are forwarded to the satellite-based emergency mapping (SEM) process, to trigger remote sensing requests as early as possible, and emergency maps production. Data products from the loss assessment and the tsunami simulations are made available to the SEM process.

Additional information on that high-level view is that scheduling groups are identified, under the diagram shown in Figure 2. This shows a red, permanent running group of processes, a dotted red event triggering computations (in blue); data from the computations are made available for a longer period through the Lexis DDI so that the desktop-based process can request access to it, and eventually ask for additional computations (inside boxes). The allocation of the tasks on the LEXIS platform is also defined, with a mix of Cloud and HPC resources for the compute parts, Cloud and desktop resources for the remaining tasks, and, of course, external resources for the systems providing data to the pilot workflow.

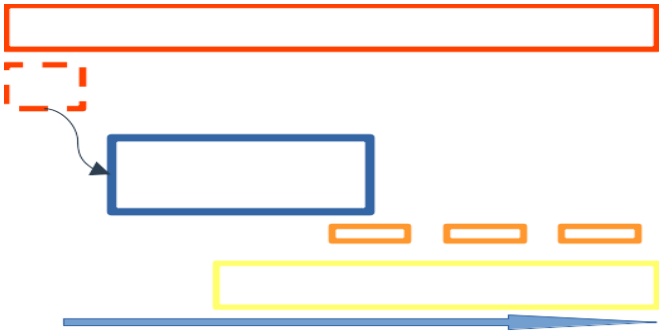

Figure 2 Scheduling groups of the workflow according to time after an event along the x axis, colours as used in Figure 1.

2.2 DETAILED DESCRIPTION

The detailed flow is represented in Figure 3, with the scheduling groups overlaid over the BPMN-like description. Individual tasks have not evolved much since the D6.1 description [12] in the overall scheme, but some of the elements have been refined and implemented (as detailed in the individual sections below). The main changes are in the areas of interest task, where it was agreed that providing multiple areas of interest (as generated by the various steps, from loss assessment to tsunami simulations) would better be provided separated (and not merged into a global area of interest). We also confirmed the need for the satellite-based emergency mapping service to have and keep an access to the data assets produced in the early stages of the flow (damage, inundation) so as to be able to refer to those on demand, with eventual calls for additional processing on those data items during that phase of the process.

The pilot consists of the following groups:

The OpenStreetMap group

This group is online all the time, processing updates from the OpenStreetMap global database every minute, and keeping the OpenBuildingMap up to date.

The early earthquake and tsunami group

This group is triggered when receiving an earthquake event and has a tight deadline to completion. It produces an estimate of the tsunami arrival time, an early inundation map, a ShakeMap and a fast loss assessment (based on aggregation at a district level or above).

The precise earthquake and tsunami group

This group is triggered after receiving the earthquake moment tensor, at around 10 minutes after the initial earthquake event. It produces precise wave height and inundation and precise damage assessments, that are used to update the information available to the SEM group.

The SEM group

This group is activated by the computation of the first area of interest produced by the early earthquake and tsunami group and is also updated when the precise group has completed. It includes some operator tasks, requiring human intervention, and produces data products (database and maps) for emergency response services.

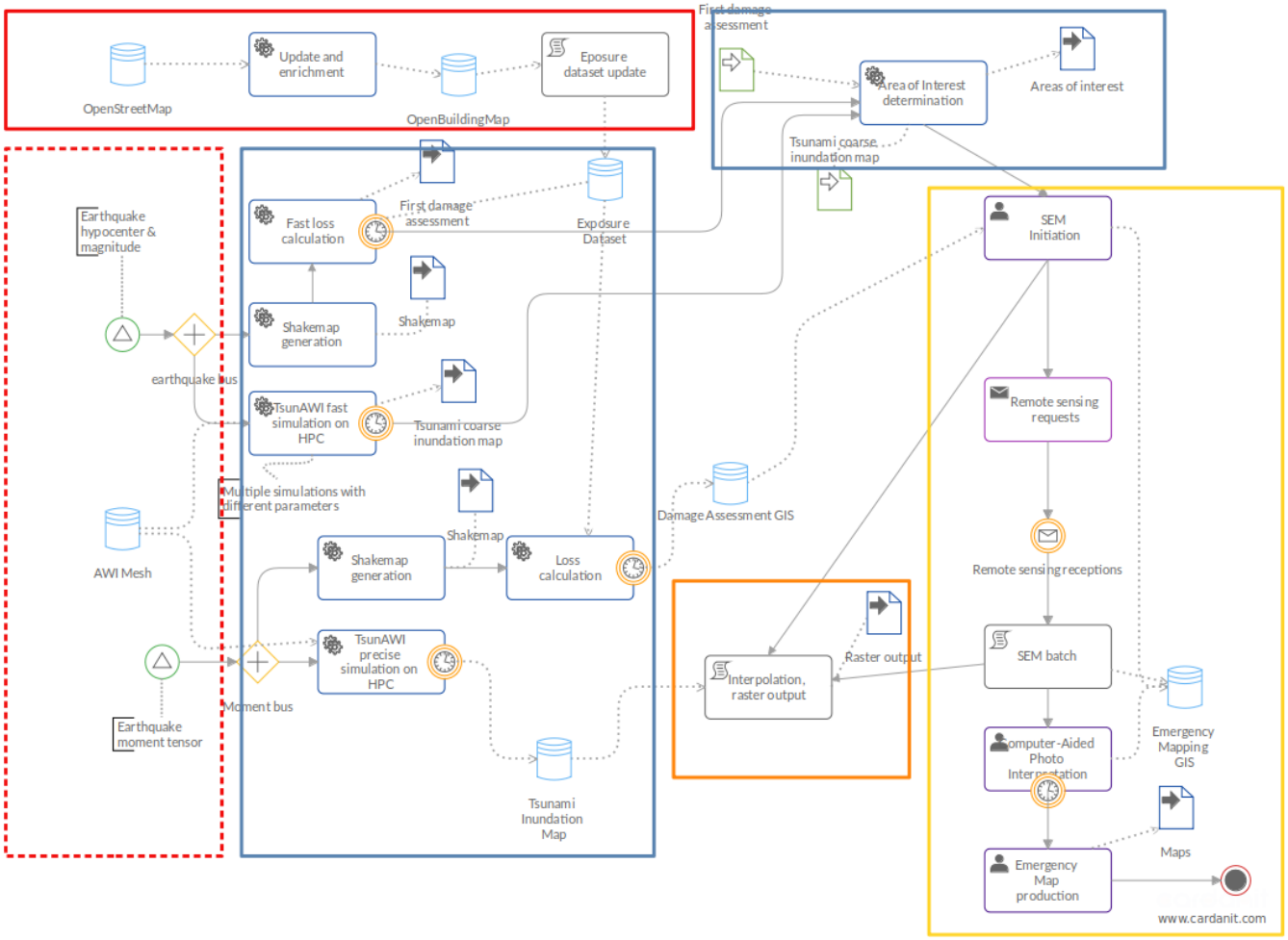

Figure 3 The detailed workflow in a BPMN-like notation, overlaid with the scheduling groups as defined above.

In the early stages, we have to take into account the uncertainties around the source of the earthquake event, and, eventually, uncertainties about resources availability on the target structures. This is answered by:

- Ensemble simulations at the fast stage with TsunAWI, with a set of input parameters based on the distribution of parameters of the initial event,
- A dispatch of multiple simulations on Cloud and HPC resources at the same time.

We are also considering the possibility of a slave run that could be used as a backup on a remote centre, so that, if a failure would happen on the main flow, a backup flow could take over, with an expected latency; but this would avoid having to restart the flow.

2.3 MODEL OF COMPUTATION FOR THE WORKFLOW ORCHESTRATION

The pilot main approach is to use a well-defined model of computation to define the workflow, and assist in its execution in the following three ways: ensure that the LEXIS orchestration properly supports the pilot, ensure that the expected latency (deadlines respect) is met, and size the required resources for effective execution.

2.3.1 Model of computation definition

The Model of Computation (MoC) used to model the workflow is a state-of-the-art data flow model called PolyGraph [3], [13]. It is based on the well-known formalism of Synchronous Data Flow (SDF), with extensions to

add a globally synchronous behaviour in addition to the SDF asynchronous message passing between components, which makes it suitable in the LEXIS context. It also supports a measure of reconfigurability at runtime for the components, which allows to differentiate system states (like for example monitoring for events and reacting to events).

PolyGraph defines an application as a set of actors (the tasks of our workflow), and a set of channels modelling asynchronous communication between actors. The behaviour of each actor regarding communication is to perform atomic transactions, called firings. When firing, an actor consumes a fixed amount of data from its input channels and produces a fixed amount of data to its output channels. The amount of data transiting on a given channel is measured in tokens. As such, every actor has a rate associated to each of its channels, defining the amount of data produced and consumed by its firings as a number of tokens.

PolyGraph allows to assign a firing frequency to a subset of actors in the model. These timed actors are then constrained to fire periodically on a common global time scale, ensuring real-time synchronization of the workflow. This defines global throughput constraints. By adding a phase to timed actors, their firings occur at different instants in the global time, defining an end-to-end latency between the firings of timed actors (detailed in [3]).

PolyGraph also allows to assign execution modes to some actors. An actor behaves in a different way depending on a current mode decided for each of its firings. The alterations in behaviour include for example different execution times, production of different data types, etc. The mechanism to decide of an actor's current mode are as follows. Some actors have the ability to produce labelled tokens, conveying a mode information to the consumers. When firing, a consumer's current mode is decided depending on the mode labels associated to the tokens it consumes on its input channels for that firing. That consumer will propagate the mode information to its successors, and as such their current mode is decided by transitivity (detailed in [13]).

Contrary to the usual behaviour of processes in data flow models, the read operation is not necessarily blocking on input channels. Some timed actors can be assigned a relaxed behaviour for their inputs. When the instant of their firing is reached (can be seen as a deadline), it fires and consumes the expected number of tokens on its input channels, regardless of their availability. This firing is actually a global transition, which cancels the firings of its predecessors in the data flow graph that did not produce their results in time, to restore a coherent state. This allows for example to have speculative execution patterns in the data flow graph.

The extended semantics of PolyGraph preserve the capability to decide of consistency (periodic behaviour in bounded memory) and liveness (absence of deadlocks) properties in SDF. By taking real-time constraints into account, it extends the notion of bounded periodic communication behaviour to bounded periodic real-time behaviour, and the notion of absence of deadlocks to time boundedness. Inheriting rich dynamic concepts from models like SADF and TPDF, it also allows to capture complex behaviour in dynamic and context dependent systems.

The use of this model in the workflow is done in the following way: first a view of the workflow as a dataflow is considered and then semantic annotations are added to refine the expected behaviour. The resulting model is illustrated in Figure 4 and explained hereafter.

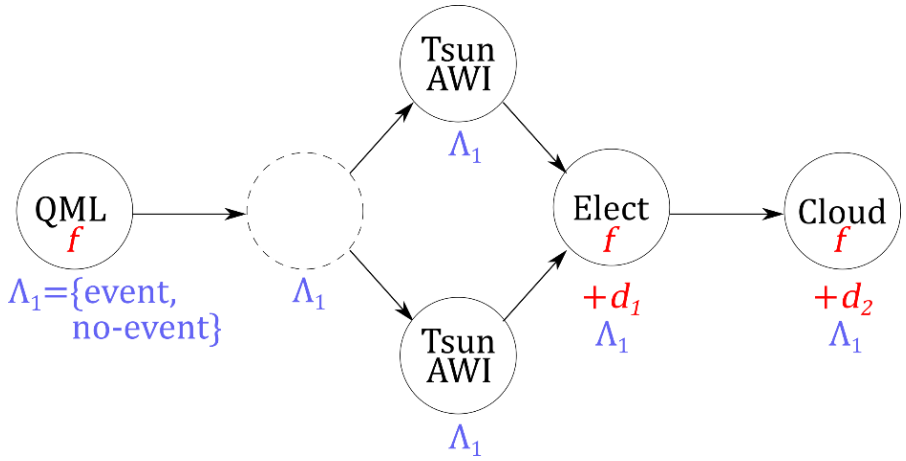

Figure 4 Coarse-grain model of a subset of the workflow; all rates are set to 1, so they are omitted from the figure.

The actor QML represents the production of a QuakeML file in case of an event. It has a frequency *f* defining the considered minimal inter-arrival time 1/*f* of an event, and the firing frequency of the QML actor. Hopefully, it will not produce an actual file every firing, since the frequency of occurrence of an earthquake should be much lower than *f*. As such, it is allowed to label the tokens it produces with one of two modes: event or no-event. When no event was detected and no QuakeML file is available, the actor produces a token labelled with no-event. Otherwise, if an actual file is produced because an event is detected, the produced token is labelled with event. This will determine the execution time of the other actors. If a no-event token is produced, we can consider that all the other actors will process in zero time for that iteration. In practice, the components will not compute and only the orchestration will be active to keep the system synchronized (state, resources, etc.). Otherwise, the flow will execute the fast simulations as follows and send the results to the cloud.

A TsunAWI fast simulation is an actor with one input channel for the tokens from QML. In the model, a dashed unnamed actor represents the sharing of this file between several instances of TsunAWI actors. There can be any number of parallel runs in the model, here we chose two arbitrarily. In practice, their number will depend on design choices driven mainly by the functional relevance to duplicate simulations (i.e. not the available resources). Each simulation will start as soon as a file is available on its input channel and resources are allocated for the run. When a simulation finishes, the corresponding actor produces its results on its output channel.

Overall, the workflow is a simple kind of dataflow, typically SDF (synchronous dataflow) where each task consumes a fixed amount of input tokens (input data such as a QuakeML file), and produces a single token of data items every time it runs (a ShakeMap, an inundation map, etc…). The difference is that with the frequency constraint on the QML actor, the pipeline is constrained to have a certain throughput. The Elect actor has the same frequency constraint *f* in Figure 4, and receives the results from the fast simulation at that same throughput. The additional annotation d_1 stipulates a time offset for the firing of the Elect actor, from the firing time of QML. This represents the first deadline, where first estimates and fast simulation results are delivered. At that time, the simulations that did not produce a result in time are cancelled, and the Elect actor chooses among the available results the most relevant, to be passed on to the next stages of the workflow. As such, we call this actor a transaction box.

The next stages of the workflow are represented here as a very coarse-grained actor, encompassing the whole cloud workflow. A second deadline d_2 is specified on that actor (also as an offset from the firing of QML), for delivering the precise results (detailed inundation and loss assessment). Estimated times for those deadlines are in the order of minutes after the event for d_1 and nearing 60 minutes after the event for d_2.

For the Elect actor, a possible feature we are studying is the ability to enhance the filtering step by taking into account updated information: for example, if the earthquake event precision increases over time, then, simulation results that were based on prior values and uncertainties could be filtered according to the more precise event

information available at decision time. This updated event information simply appears as additional communication channels as input to the Elect actor, and, as per PolyGraph semantics, can handle both availability of updated information or no availability.

2.3.2 Mapping to the LEXIS orchestration resources

The second aspect of PolyGraph for the pilot is to consider the mapping onto the infrastructure of the LEXIS project; that is how PolyGraph is implemented over that infrastructure.

Considering that the project provides a unified data and orchestration infrastructure, a first view of PolyGraph mapping was to build a meta-task for the pilot, that would manipulate the orchestration and data layers, start jobs, monitor their completion, and track time to properly wake up the deadline triggered tasks. This could be done via events sent by the orchestration layer, and an API that would allow the meta-task to know in which orchestration context it is run, and so which target must be used to manipulate the flow. The initial idea was then to build a messaging service to keep track of that.

Discussions with the technical work packages, especially during the September 2019 F2F meeting, pointed out that this approach was unrealistic, in part because one of the sub-orchestration layer does not provide a way for jobs to know in which context they are run.

The answer was then to consider what would be needed from the orchestration to support the model, and it turned out to be fairly simple: deadlines in the orchestration. Typically, an orchestrator provides for time-outs, for example when submitting a job, and on-error or on-timeout decisions (like starting a different job if the timeout is triggered). Such flows allow for example, submitting a job with a timeout, and, upon the timeout trigger, cancel the submission and submit another job. However, timeouts are not like deadlines: in our workflow, it is essential to trigger behaviours or tasks upon time to completion violations, and this covers both a timeout upon submission and timeout for completion of a job, except that timeouts are not equivalent to a deadline.

The resulting decision was to introduce the support of deadlines in the orchestration, and the use of those deadlines by PolyGraph; PolyGraph tasks will then appear as jobs in the LEXIS orchestration, and there would be no need for a meta-task and the use of an API to ensure the completion of the MoC-defined workflow.

The second aspect of PolyGraph is the scheduling based on data availability (i.e. tokens). This can be handled at the orchestration and data layer levels of LEXIS directly; triggering jobs on data availability is doable, and may require specific data tasks that ensure that results produced by a job are made available to a successor job. This is a common approach to model and integrate data movements in dataflow runtimes, and we are familiar with that approach.

2.3.3 Latency and resources determination

The reason for the model of computation used is the capability to determine time related quantities on instances of it, such as latency and throughput. This is important because it allows to determine, given timing information on the tasks of the workflow, whether deadlines can be met at all or not, and the necessary resources (minimal set) to do so, and the set of resources needed to be able to meet a period criteria, which is the delay before a flow can be started after the start of current one (obviously, if that delay is greater than the completion time of the flow, this is always possible).

The first question, about latencies, is made easier by the specifics of the MoC we use; the presence of a transaction box means that the latency has to be met by at least one of the equivalent incoming connections to the transaction box and not all of them, strictly reducing the requirements that the flow has to meet. For example, if we decide to run ensemble simulations of the fast TsunAWI simulations, then it becomes enough that only one of those complete to meet the deadline; any additional simulation comes as an enhancement for the situation knowledge available at that point.

The second question is a sustainability criteria: for this kind of process, destined to be started at any point in time in cases of emergency, then guaranteeing that resources are always available means being able to size said resources and therefore to bill for the minimum set, with an appropriate set-up (this subject will be elaborated in the exploitation part of the project). And, the delay (linked to the throughput) is part of that evaluation, since guaranteeing that the processing chain can handle another event within a fixed delay suppose also that we know how many resources have to be safeguarded to meet that requirement.

The methodology for those evaluations becomes the following: additional tasks and connections are added to the instance defined above, to be able to represent additional timing information. In particular, we can model set-up times for certain compute tasks by considering an orchestration job to be a pipeline of two tasks: a set-up task, which, once completed, sends a token to the compute task. In that way, we can model the overlap of compute and data movements, such as starting the set-up in advance (before, for example, the data is available) and having a ready to compute task that is waiting for a data availability afterwards. Such a set-up task may only be an abstraction and may be used only to be able to compute latencies, without real existence in the workflow.

In the same abstract way, a delay requirement is captured as a dependency between the end and the start of the workflow; by preloading it with tokens, we can express that delay into the number of flows that can be started during the execution of the first flow.

3 OPENBUILDINGMAP

3.1 OPENBUILDINGMAP UPDATE

3.1.1 Processing optimization

To ensure fast performance of the OpenBuildingMap in delivering exposure data for rapid loss assessments and to keep the exposure database as small as possible, we have adopted a new schema for pre-processed exposure data. The definition of this schema was driven by the fact that exposure data needs to be delivered in various forms:

1. On a building-by-building level if the area of interest has complete building coverage and if privacy laws allow the distribution (e.g. to civil defence authorities).
2. On an aggregated level per area if the building coverage is incomplete or if privacy laws do not permit dissemination on the building level.
3. On a mixed level for areas with incomplete building coverage if privacy laws permit the detailed dissemination. On this level, we provide building information for the buildings in the database and aggregated information for the missing buildings in the database. While this level certainly constitutes a compromise, the whole system gradually converges to the building-by-building level with increasing completeness.

The new schema employs the classical quadtree approach [14] as used by the most map services on the Internet. This approach basically increases the number of square cells (tiles) from zoom level 0 (one cell) to each subsequent zoom level by splitting each cell into four equally sized subcells. The target zoom level for exposure data is 18 (up to 150m side length) with more than 68 billion cells if the Earth were to cover completely with cells at this zoom level. However, due to the quadtree approach, we can reduce the resolution by combining cell groups into lower zoom-level cells that cover water or empty areas (no buildings at all) and thereby significantly reduce the number of cells in the database.

The procedures to combine cells or split cells into the upper or lower zoom level is implemented. The next steps will include handling the payload per cell in the combining/splitting processes. The payload per cell are the aggregated exposure files, the remote-sensing polygons, and the exposure data per building within the respective cell. The current concept foresees the payload to be stored in a directory tree resembling the quadtree structure. Through this mechanism, delivery of precomputed exposure data per cell or polygon of interest should be by far faster than accessing the OpenBuildingMap database through expensive geographic queries.

3.1.2 Cell completeness estimates

As described above, the form of dissemination of exposure data as well as the resolution on which cells are stored depends on their completeness in terms of building coverage. We distinguish between complete cells that contain all buildings, almost complete cells in which only one or a few irrelevant buildings are missing, incomplete cells in which all or some buildings are missing, empty cell in which no building exists, water cells that cover only water, undecidable cells for which we cannot determine the ground truth because of e.g. blurred satellite imagery or cloud cover, and finally cells with unknown status. To create these completeness data, we are following a multi-tier approach:

- We built a website on which an analyst can set the completeness status for one or more cells by comparing the satellite imagery with the overlay of building polygons directly. Figure 5 shows the previous version using a 10-arcsecond grid instead the new quadtree cells.

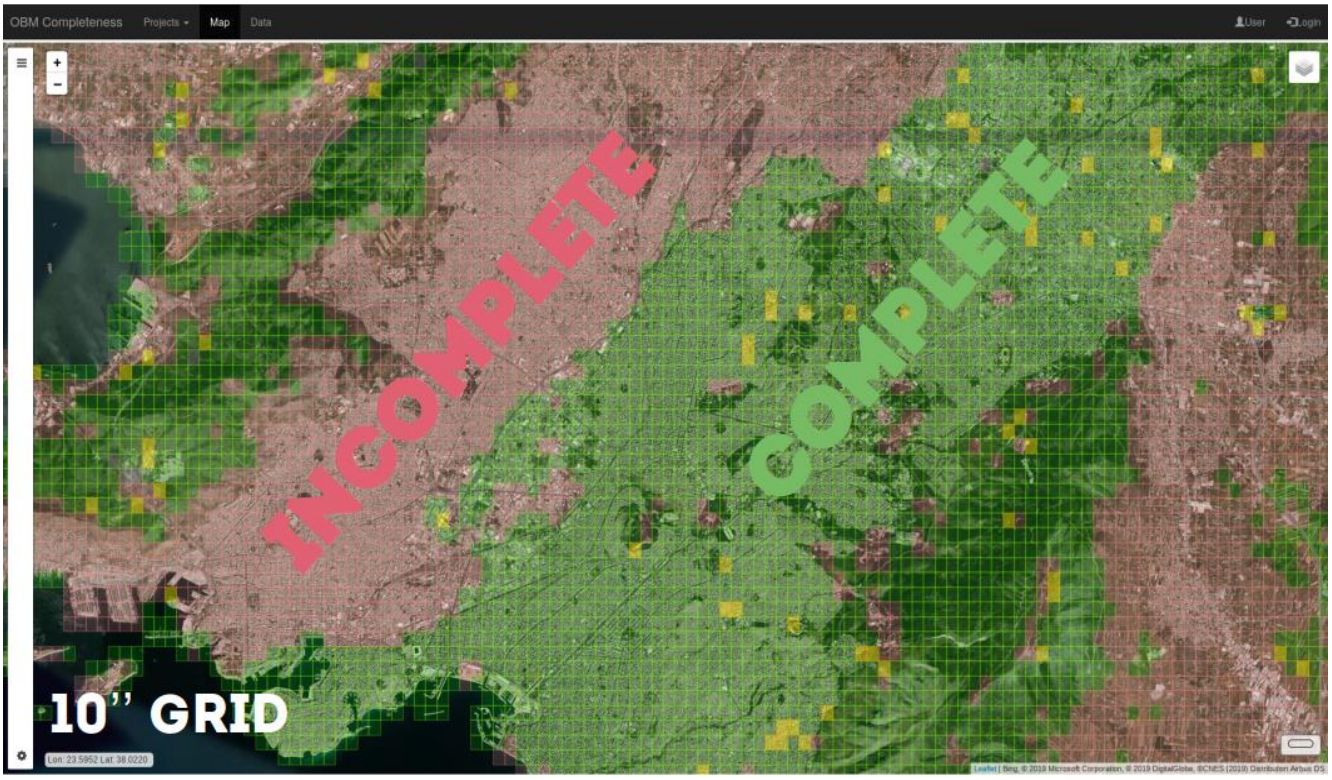

Figure 5 Cell completeness estimates (previous version using a 10-arcsecond grid instead the new quadtree cells). Screenshot of the website to set the completeness status for one or more cells by comparing the satellite imagery with the overlay of building polygons.

- We have started a collaboration with the team at the University Heidelberg/HeiGiT to include a completeness analysis into the *MapSwipe* smartphone application. *MapSwipe* has been developed together with the Humanitarian OpenStreetMap Team and is crowd-sourcing the detection of buildings from satellite imagery. The users are presented satellite imagery at the zoom level 18 and they have to decide whether or not an imagery tile contains buildings. This selection is later used to provide people tracing buildings from satellite imagery with a preselection of areas where buildings are to be found. *MapSwipe* has been extended to be able to now overlay vector data (in our case building footprints) and have the user confirm whether or not the footprints match the satellite imagery, delivering a completeness estimate per tile/cell.

- Because *MapSwipe* results so far are publicly available, we can use this data to identify two types of cells: empty cells and incomplete cells. Empty cells (no building at all or water) are the cells not marked by the users as containing buildings at all. Overlaying them with the global coastline, we can distinguish empty cells from water cells. Cells can be interpreted as incomplete if they are marked by the users as containing buildings but no building footprint is present. This approach will not necessarily deliver a completeness classification for each cell as they are undefined states, but it will help us to populate our database with completeness information.

- The last approach is less accurate but can be employed automatically and globally. With remote sensing techniques based on mainly Sentinel-I radar data, we can identify built area. Preliminary investigations in areas with complete building stock data show useful correlations between the size of the built area and the size of the building footprints such that in absence of more detailed information, the remote-sensing estimate of built area can be used as proxy to number of buildings per cell.

3.1.3 New processing steps

As described above, exposure data will be provided on a per cell (preferably zoom level 18) basis, independently on the form of dissemination. Therefore, exposure information of each building is stored in the directory of the respective cell together with the aggregated exposure as given by outside experts. In the case of Europe, we are currently importing the SERA exposure data as distributed by the Global Earthquake Model. This data is organized by districts with different administrative levels depending on the country. Each district's dataset is distributed proportionally (to the portion of each cell within the district) to the cells. If remote-sensing information for a particular district is available, the exposure will be distributed according to the proportions of the built area per cell and not the total cell size, thereby increasing the resolution of building and population distribution even if building data is not available. At this stage, exposure data can be quickly disseminated on the cell level as it is readily available. Things are becoming more complex when buildings are present. In that case, each building will be separately assessed, and its exposure information will be stored in separate files in the cell directory to which the building belongs. With any change to a cell, the cell will be completely reassessed, and overall exposure will be computed considering the available building information and the basic exposure information for the cell. Depending on the completeness level of the cell, this information will be combined preserving the highest possible level of detail. Figure 6 shows a set of the previously used 10-arcsecond cells.

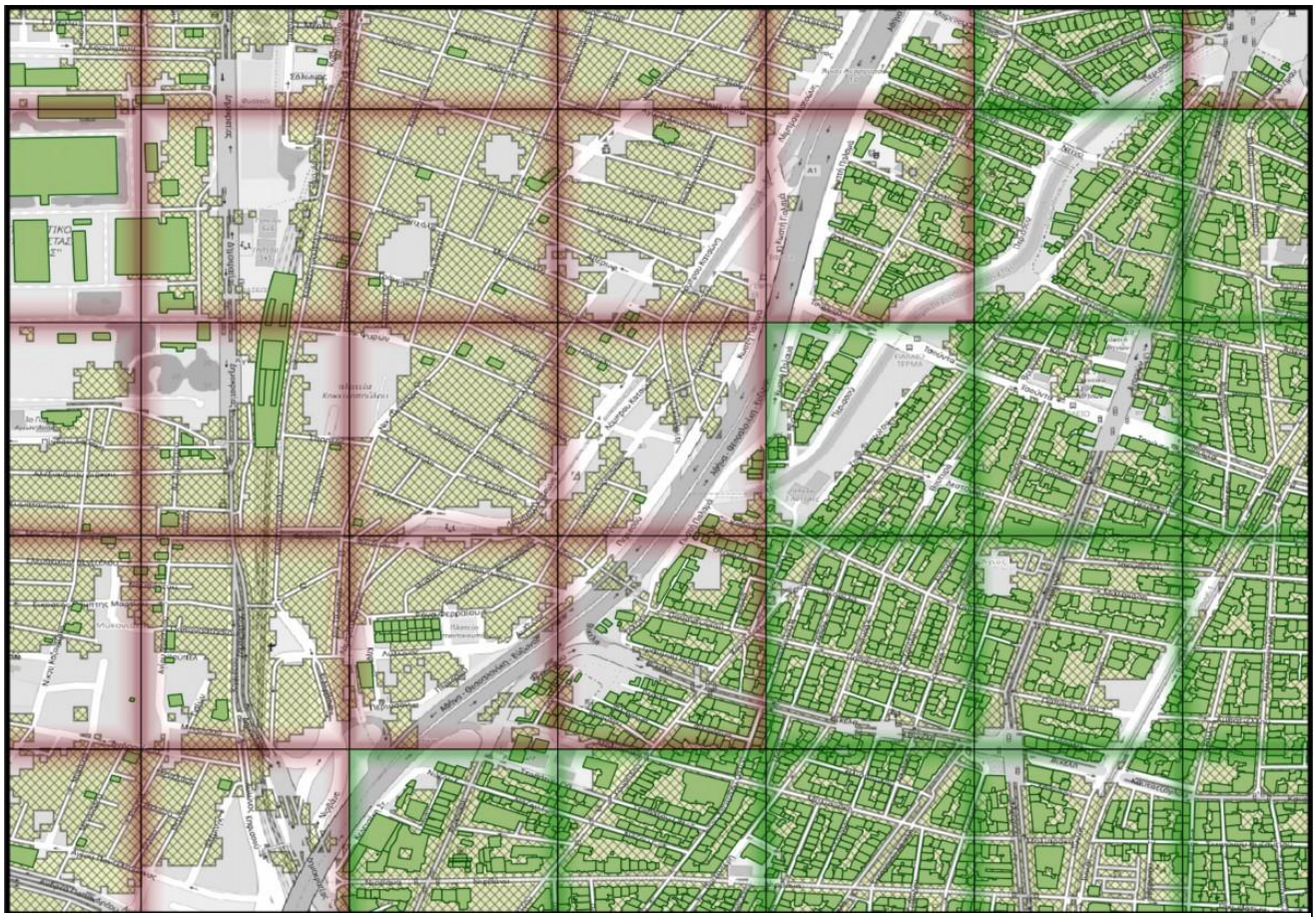

Figure 6 Cell completeness estimates - completeness level of cells (new quadtree cells)
Green-framed cells contain the complete building stock while red-framed only cover incomplete building data. Yellowish polygons indicate the built area as detected by remote sensing and green polygons represent the available building footprints.

3.1.4 Rule-based DSL

In the OpenBuildingMap process, a significant step is enriching the building data obtained from OpenStreetMap with additional data and knowledge, following two steps:

- Geometric information, either computed from the original objects and shapes, and area-specific knowledge
- Semantic information on the building function and specifics, up to the point where the building vulnerability function can be established

This process relies on the integration of domain knowledge, from multiple domains (from geographical information systems to civil engineering and disaster) into scripts that can be run against the original data and recalled for every update of the data set.

This process is computationally intensive, reaching a limit to the number of updated buildings that can be processed by time period, and also increasing the time needed to rebuild the database out of an OpenStreetMap snapshot. For example, considering that in early 2020, the number of buildings in OpenStreetMap was about 375 million, and that a system could process 1 million buildings per day, it would take about a year to rebuild the database.

A second issue is the integration of domain knowledge. If scripted, then the introduction of a new rule would mean looking into the existing scripts, finding the appropriate one, and coding the rule inside the scripts, paying a close attention to the rule orderings; rules having the characteristics of depending on data computed by other rules needs to be executed after all the previous rules have been processed properly.

A studied improvement for that context is two-fold:

- Consider that the process described above can be seen as a two-steps knowledge database process, with an extensional database (EDB [15]) containing the facts retrieved from OpenStreetMap, and an intensional database (IDB [15]) containing the data inferred by the scripts.
- And describe by declarative rules the domain knowledge brought by the experts in the original scripts. The original scripts would be then removed, and the IDB would be created out of those rules.

This would help cover the following issues:

- The ordering of rules activations would be solved by the database engine:
    - Domain rules can be processed so as to extract their dependencies (i.e. track data items used by a rule, tell whether those are in the EDB or IDB, and, for those in the IDB, find which rules are in charge of inferring those data items).
    - From the dependencies, build an ordering of the possible activation sequence of the rules. Typically, we will obtain a partial ordering, and hence have multiple possible orderings of the rule activation. Also, when building that partial ordering, we would also discover if we have cycles between rules (a naive example being the rule computing B is dependent on A, which is computed by a rule itself dependent on B).
- Interaction with the database would become set-oriented, increasing very significantly performance by processing by objects sets in a single transaction. For example, when activating a rule, select all objects that matches the rule preconditions, modify all those objects with the rule inferred data items, and write them back to the database in a single transaction.

The implementation of that scheme was discussed, and, however, was evaluated as problematic. OpenBuildingMap relies on an open source database engine (PostgreSQL) with geographic extensions (PostGIS), as well as the import open source tool osm2pgsql. The scheme described in Figure 6 is typical of a deductive database [15]; finding an open source deductive database with geographic extensions and the same scalability and performance as

PostgreSQL is problematic. We therefore have started to discuss with European innovators in the database space so that they can use the LEXIS portal to experiment and see if they can propose us a solution or a path towards one.

3.1.5 ShakeMap generation

We developed a web engine to generate synthetic ShakeMaps harnessing the OpenQuake engine of the Global Earthquake Model (GEM) foundation. The back-end asynchronously digests requests parameterizing earthquake sources in terms of source depth, epicentral location, moment magnitude and focal mechanism. The back-end returns shaking in user definable ground-motion measures (e.g. PGA or IMS) and can be retrieved in various formats such as ASCII, GeoJSON, among others. This tool implements an open and documented API that users and other services can query systematically and automatically. An interactive interface allows to explore the expected spatial shaking distribution by selecting locations of interest on a map and defining the earthquake source interactively in a web browser. An example of the query using the interactive web browser front end is shown in Figure 7. Besides the interactive mode, this service now provides, through HTTP requests, a simple interface for any type of ShakeMap to be used in automated systems that require rapid ShakeMap computations without the need to run local instances of OpenQuake.

This tool is used within RISE to test, validate, and benchmark the effect of ground motion prediction equations (GMPE) on loss assessment scenarios (freely available[1]).

A repository has been created to host representative scenarios which have been designed and made available to all project partners to ensure efficient streamlining of development as well as compatibility during the development phase[2].

Currently, the front end is being extended by a quantification and visualization of exposed buildings for a given intensity range using open building data bases overlain by the calculated intensity maps.

The current stable implementation exposes all available ground motion prediction equations to the user. All of these use the commonly used Vs30 (the time-averaged shear-wave velocity to 30 m depth) models as site condition proxy (SCP) to derive the ground motion. As a next step the Vs30 SCP will be extended by the slope derived from digital elevation models (DEM). This allows to improve the accuracy of local site amplification predictions. This step will be complemented by a region specific GMPE recommender which will select the best suited GMPE for a given site selection.

[1] shakemAPI: https://gitext.gfz-potsdam.de/marius/ShakeMapi

[2] Repository with representative scenarios: https://gitext.gfz-potsdam.de/marius/lexis-scenario-examples

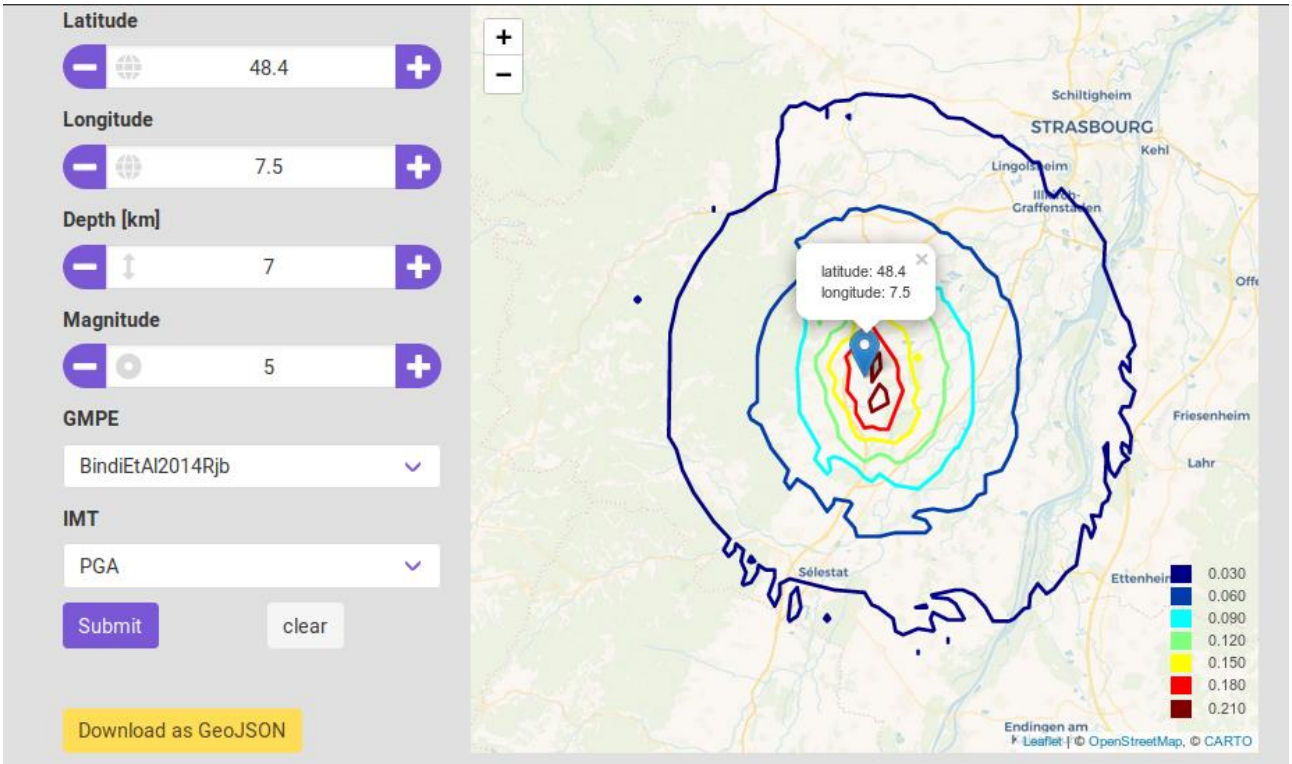

Figure 7 Example query using the interactive web browser front end. Colours indicate PGA.

3.2 DAMAGE AND LOSS ASSESSMENT

Damage assessment is technically the combination of the ShakeMaps with exposure data enriched by fragility functions. This part of the computational chain is the least expensive one as long as the ShakeMap and the respective exposure data is available and in the correct format for the OpenQuake processing engine. Preliminary tests have shown that damage and loss assessments can be computed within minutes on regular desktop computers in a single thread. Therefore, we do not anticipate this part of the pilot to become the bottleneck. Efforts are rather concentrated on making the exposure model readily available through precomputed exposure files for OpenQuake and efficient file discovery when searching for exposure of a given region of interest.

4 TSUNAWI

4.1 BENCHMARKS

4.1.1 Padang inundation on fine and coarse mesh

Within the earthquake (EQ) and tsunami work flow, a rough estimate of the tsunami inundation has to be delivered fast, followed by a more precise simulation later. The precise simulation for Padang, Sumatra, Indonesia, was provided in D6.1 [12], and for D6.2, we added a set up on a coarse mesh. A sensitivity study with varying mesh sizes lead to the conclusion that a resolution of 200m on land still retains the most important patterns of the inundation, while reducing the computation time by a factor of 60 compared to the original set-up with 20m resolution on land. In particular, in most regions, the inundation is slightly overestimated in the coarse simulation, with minor exceptions of local valley-like features only resolved in the fine mesh. An example is shown in Figure 8.

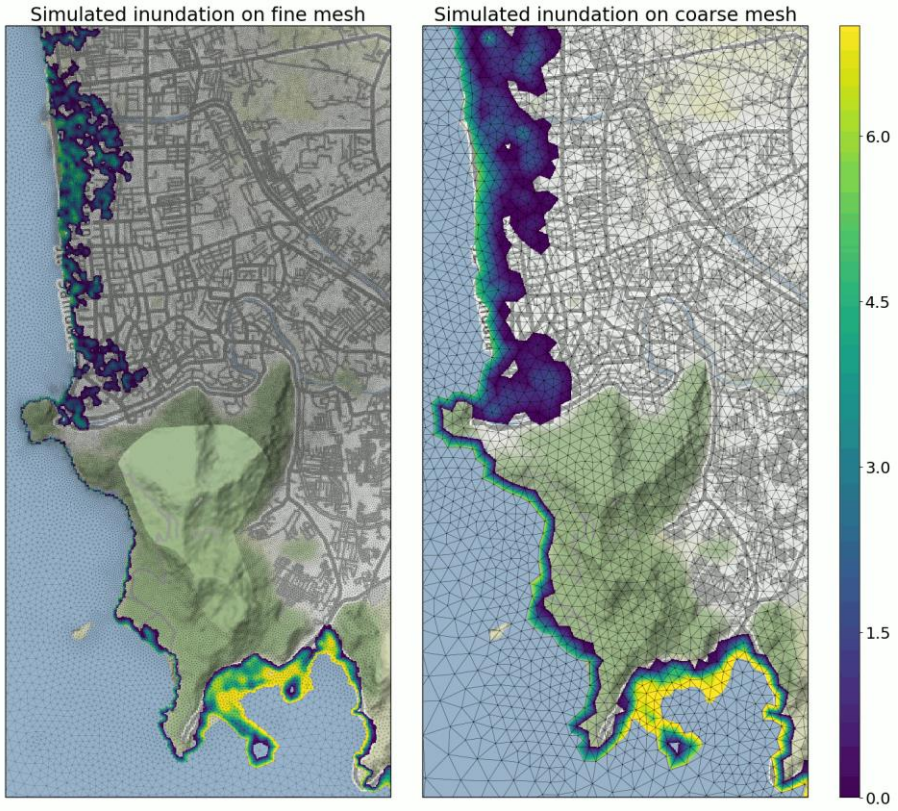

Figure 8 Simulated inundation [m] of a tsunami caused by a hypothetical Earthquake of magnitude Mw=8.8 west off Padang.
Left: with a high-resolution computational mesh; Right: low resolution.

In addition to the new coarse simulation, we could improve the computation time by increasing the time step. Choosing the largest possible time step requires to simulate tsunamis caused by different high magnitude EQs, to ensure stable simulations. The table below shows some figures on the two benchmark cases for Padang, together with computation times on different HPC systems. In particular, the speed up on the very recent installation "Lise" of the North German Supercomputing Alliance (*Norddeutscher Verbund zur Förderung des Hoch- und Höchstleistungsrechnens* – HLRN[3]).

[3] HLRN-IV Systém: https://www.hlrn.de/supercomputer-e/hlrn-iv-system/?lang=en

		DETAILED MESH	COARSE MESH
NUMBER OF MESH VERTICES		1,242,653	231,586
RESOLUTION	in Padang in the ocean	20m 5,000m	200m 15,000m
TIMESTEP		0.15s	1.5s
COMPUTE TIME for a 2h **SIMULATION**	salomon.it4i.cz, 24 threads, 2x Intel Xeon E5-2680v3 ollie.awi.de, 36 threads, 2x Intel Xeon E5-2697v4 lise.hlrn.de, 192 threads, 2x Intel Xeon Platinum 9242, with hyperthreading	20:40min 15:45min 5:04min	20s 15s 5s

Table 1 Benchmark cases for Padang, together with computation times on different HPC systems.

4.1.2 Chile 2015 earthquake and tsunami

As the Padang test case is an artificial benchmark, we agreed to add a test case based on a real event with a good coverage by observation and measurement. The Chile 2015 earthquake and tsunami is a good candidate. On 16 September 2015, the earthquake of magnitude 8.3 off the coast of the Coquimbo region caused a tsunami

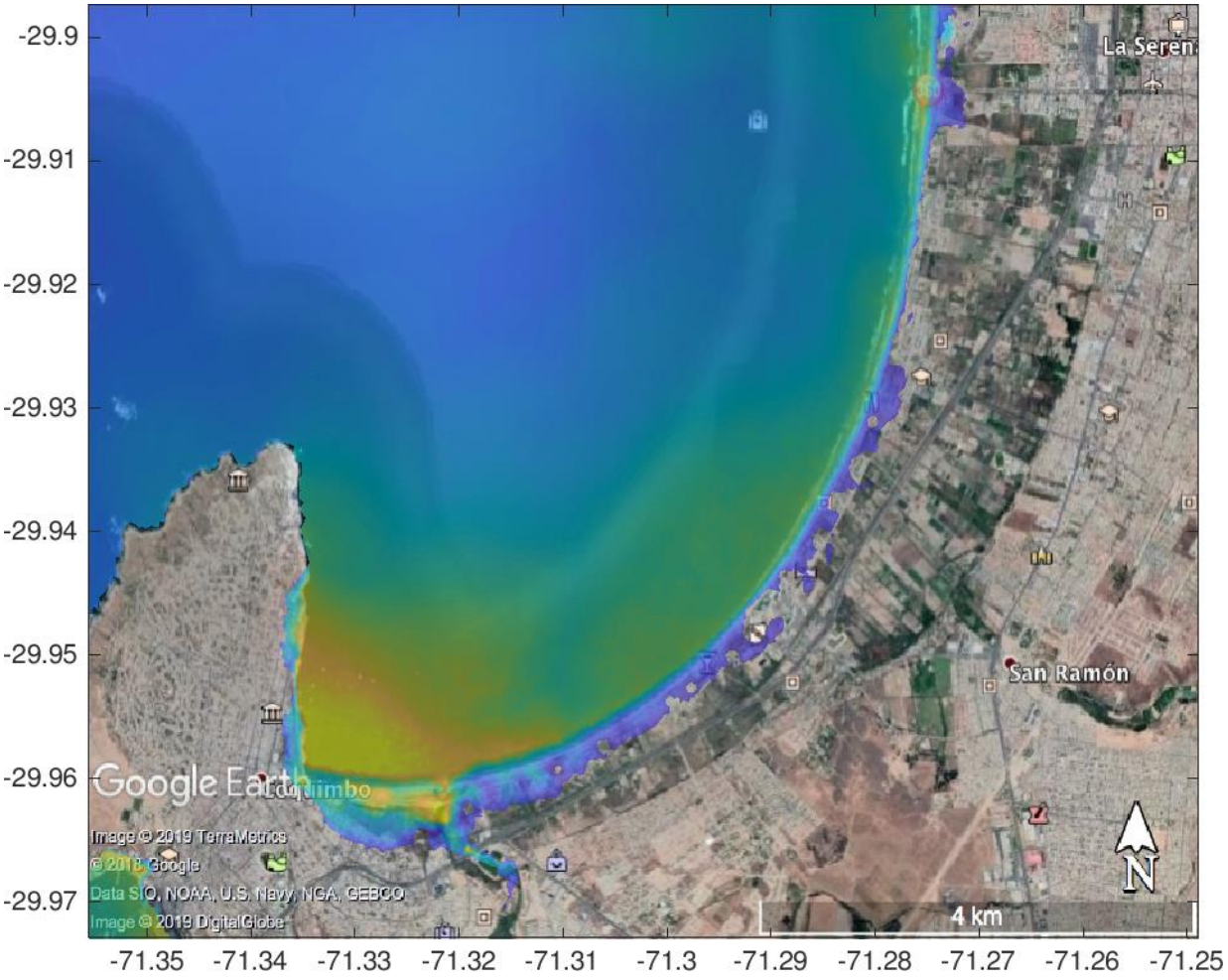

Figure 9 Simulated inundation of the 2015 Coquimbo tsunami, Chile.

with significant inundation in the port of Coquimbo. Within the project RIESGOS (Multi-risk analysis and information system components for the Andes region), funded by the German Federal Ministry of Education and Research, we

have access to high quality topography data provided by Chilean partners. Based on these data, the inundation can be simulated very well with TsunAWI, as depicted in the figure below.

However, for the use in LEXIS, we must rely on other data sources, but the results are far less realistic with free SRTM (Shuttle Radar Topography Mission) data. Maybe, we have to acquire commercial topography data for the region of interest. As a first step, we have to agree with the project partners on the area to cover with simulations. Second, we need a set of initial conditions for the fast simulation, before the momentum tensor is available. Again, Chilean partners provided sources for restricted use in RIESGOS, and we have to either fall back to simple Cosine bells or investigate on other sources with geographic information on the trench. In first tests, Cosine bells result in simulated wave heights about twice as high as those simulated with the RIESGOS sources.

4.2 SOURCE CODE IMPROVEMENTS

4.2.1 Additional input options

In past projects, TsunAWI was only used in offline mode to set up a database. To ease online computations, we have added input from the command line complimentary to the Fortran namelist. It is possible to pass the EQ epicentre, the magnitude, a scenario ID, or a path to a QuakeML file.

If given, the QuakeML file is parsed for EQ epicentre and magnitude. As a next step, the momentum tensor information will be parsed and the initial conditions for the tsunami have to be derived. However, this step has still to be implemented.

To choose the kind of initial conditions on the fly became an additional task. About 10-15min after an EQ event, the moment tensor becomes available with information on the geometry of the rupture. Before, we have to rely on pre-computed initial conditions, as available for the Sunda Arc offshore Padang, on Okada parameters pre-determined for the fault, or as a fall back on an idealized source in regions without any pre knowledge. If TsunAWI is started with the EQ coordinates without further information on the source, it now checks for the availability of a predefined source for the EQ location and falls back to a Cosine bell scaled according to the magnitude if no precomputed initial conditions or Okada parameters available.

4.2.2 Interpolation to raster data

TsunAWI offers the option to write sea surface height, estimated time of arrival, and the maximum amplitude to a GoldenSurfer Grid raster file. The interpolation scheme is a linear interpolation from the vertices in the irregular triangular grid to the pixels in the raster. This used to be sufficient, because the raster data was used for visualisation only, while the warning products like the wave height and arrival at the coast or the inundation were derived in additional post processing steps. Within the LEXIS workflow, however, the raster data of the maximum inundation height on land is the major data product and a simple linear interpolation might result in a drastic underestimation, in particular if each pixel in a coarse raster covers a larger area.

Therefore, we added an interpolation algorithm that extracts the maximum value from all mesh vertices in each pixel. If the resolution of the mesh is similar to or coarser than the raster resolution, this approach produces some empty pixels, which in turn are filled with a linear interpolation based on the triangular element that contains the centre of the pixel.

The workflow is planning to export that part to a separate module, so that multiple raster queries can be made on-demand (different resolutions). It is presupposed that acceleration of the interpolation to raster can be done, typically on the burst buffers of the project. Maybe some of the tools of the GRASS set (Geographical Resource Analysis Support System) such as v.surf.idw or v.surf.rst can be used, based on a point output of TsunAWI.

4.2.3 Optimisation

TsunAWI has been developed for 15 years with one emphasis on computational efficiency, and the profiling performed in LEXIS revealed no low hanging fruits for further optimisation. The code is not a promising candidate for FPGAs, because this would require a major recoding from Fortran to C, and the expected benefits are small because of the data access patterns induced by the irregular triangular mesh. Similar considerations, high recoding efforts versus low expected benefit, also hold for GPUs. Nevertheless, there is plenty of room for improvements!

Load inbalance

One major issue is the load imbalance induced by different computations to be performed depending on if a value belongs to a vertex, element, or edge already reached by the tsunami or not. In addition, the extrapolation scheme in the wetting-and-drying-zone is more costly than the straight forward computation according in deep water.

A simple way out would be dynamic scheduling of the OpenMP parallelised loops, but at the price to spoil the data locality. If TsunAWI is run on a single socket, it is advisable to switch on dynamic scheduling, but in typical dual socket compute nodes, the access to data on the remote socket is too costly. Another way out would be smaller chunks combined with the default static scheduling. First tests looked promising, with a gain of about 10% wall time compared to default chunk sizes, but tests on more recent CPUs with a larger number of cores and thus OpenMP threads resulted in unstable behaviour and segmentation faults.

Furthermore, there are many vertices on land that are never reached. The computational mesh is built with clipping rules to remove vertices beyond a certain distance to the coast (typically 10km) and height (500m). These thresholds are chosen such that even very strong tsunamis remain inside the mesh, with the result that many vertices are never reached. We will simulate an ensemble of very strong tsunamis and clean the mesh from those vertices.

Data Access Patterns

Compared to Cartesian grids, irregular meshes used in TsunAWI have the huge advantage to better represent geometries like coastlines, and they allow to seamlessly change the resolution depending on criteria like water depth or region of interest. This flexibility comes at a price of more complex data structures, with indirect addressing and a higher risk of cache misses.

The computational mesh is resorted along a space filling Hilbert curve (SFC), which results in a good memory layout on all levels of the memory hierarchy from L1 cache to OpenMP chunks [16]. On each level, even better distributions are possible by other methods. In particular, from the domain decomposition point of view, the interfaces are not as smooth as they could be. But the mesh is resorted only once, directly after construction, and then fits well on all kind of many-core architecture.

Analysis of the space filling curve and data access pattern was undertaken, with a few tools and perspectives for further analysis and maybe additional tools. A computational optimisation in the TsunAWI code is the i_wet array, that is used to distinguish between vertices that require computation, and vertices that do not require computation at a given iteration. Working out of the Padang 200m resolution mesh, we mapped, per iteration, this array (completed with neighbouring information over i_wet for each vertex), over a 2D space following a Hilbert space filling curve (with the mesh traversal being interpolated over the closest higher index Hilbert curve), producing such an image as in Figure 10. This shows a significant computational imbalance illustrated by the clear and white areas, especially at the end of the iterations (bottom right corner of the figure).

Further analysis over the 120 iterations of the simulation showed that the i_wet array changes very little during the run, which indicates a few possibilities for optimisation and acceleration: first, it turns out memory access patterns are not random, but, to the contrary, streaming the values for the neighbouring vertices in advance seems perfectly doable, along with the stream of vertices to compute. Secondly, that the i_wet array does not change over iterations may allow effective scheduling strategies to be planned beforehand, or that the mesh could be optimised. An

example is given in Figure 10, where the mesh has been optimised (by removing about 30% of the vertices since in no scenario they could be reached) and the load imbalance so corrected.

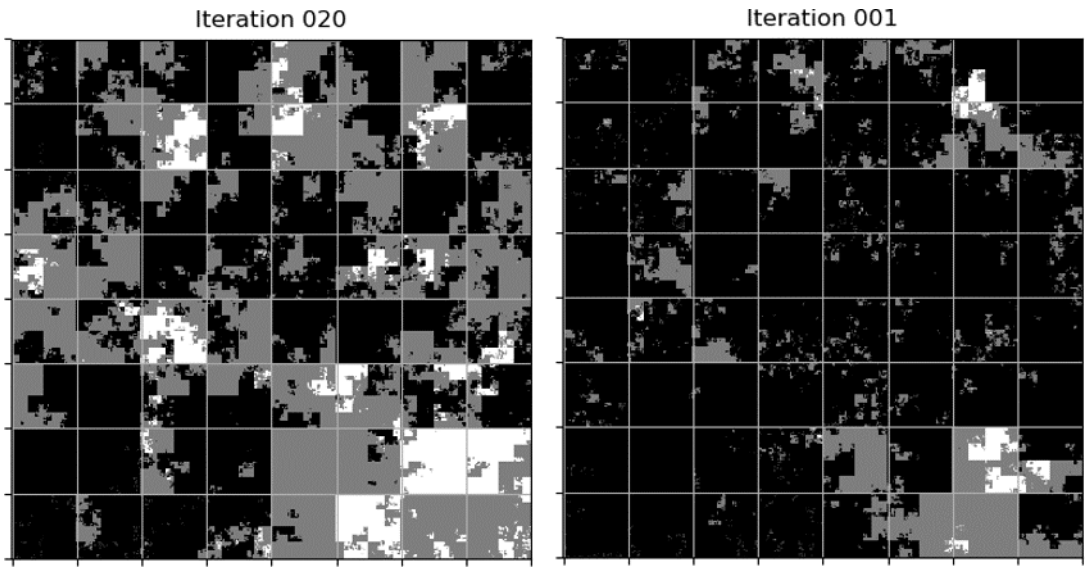

Figure 10 TsunAWI optimisation – Padang 200m mesh.
Left: Padang 200m initial mesh at iteration 20, showing the repartition of the estimated computational workload along the mesh, the traversal being mapped over a Hilbert space filling curve. Black means maximum compute, white means no or little computation.
Right: Padang 200m reduced mesh i_wet heatmap at iteration 1, showing the repartition of the estimated computational workload along the mesh, the traversal being mapped over a Hilbert space filling curve. Black means maximum compute, white means no or little computation.

Among the interesting possibilities this study is showing so far are adaptive scheduling of the loops, so that imbalances can be compensated, as long as the locality properties are maintained (smaller chunks spread over multiple cores may increase cross-core accesses at the interface between chunks, even if the SFC is trying to make those interfaces as small as possible). Another one is to have scheduling on heterogeneous targets, considering that the i_wet array allows us to properly estimate the computational and memory density associated with a chunk of the mesh, and, for example, massively parallel accelerators may be better suited for chunks predicted to have a high computational load (and a high computational load means that both mesh vertices and their neighbours can be streamed to a device).

Further Optimisation Options

In TsunAWI, the OpenMP regions are opened and closed frequently, because the attempts to reduce the number of OpenMP regions used to result in code crashes. According to conversations on Intel performance workshops, this could have been due a bug in Intel OpenMP which is fixed in the recent compiler versions and it is time to give it a new try.

As TsunAWI on many-core systems is memory bandwidth bound, a change from double to single precision can increase the performance considerably. First tests show that a speed up of 30% can be achieved with a complete switch to 4byte real size and arithmetic, still retaining reasonable results. However, it needs more careful investigation where 4byte precision is sufficient.

Another interesting idea addresses ensemble runs. TsunAWI could be rewritten to simulate, say, 8 or 16 tsunamis within one run. This would reduce the overhead of the set up and of the handling of the unstructured data access patterns within each compute loop. With a simple implementation of the numerical kernels, 8 ensembles could be calculated in the time of 6 separate runs. With some optimisation, the throughput could become even better, and

we will keep this idea in mind when we have to fill a data base of tsunami scenarios again. Within LEXIS, time to solution of each single scenario is the key figure.

4.3 TAILORING FOR THE LEXIS WORKFLOW

As described in Section 4.2.1, command line parameters, parsing of QuakeML files, and an automated choice of the best available source for given EQ parameters were added to TsunAWI. Furthermore, TsunAWI now delivers an error code on exit: “STOP 0” for successful run, “STOP 1” otherwise.

A very important aspect is the concise description of the quality of the simulation, in particular

The quality of the EQ source: idealised Cosine bell (very low), pre-knowledge on the fault (medium), momentum tensor of the actual EQ (best),

- The mesh resolution: fast estimate of the inundation on the coarse mesh (low), accurate estimate on the fine mesh (best).

As suggested by ITHACA, we will add this meta data according to ISO 19115 “Geographic Information - Metadata”.

5 SEM

The main goal of the Satellite-based Emergency Mapping (SEM) system is to supply information for emergency response during and after a disaster event.

The Copernicus Emergency Management Service (CEMS), the European Commission SEM service, is based on two main components: Early Warning and Monitoring, and On-demand Mapping services. The On-demand Mapping services provide rapid maps for emergency response and risk & recovery maps in support of disaster management activities not related to immediate response.

The CEMS Rapid Mapping service is based on satellite images and geo-spatial data in order to provide a mapping service in case of natural disasters and human-made emergencies within hours or days immediately after the disaster event. With its 24/7 availability, the service covers all the phases of the emergency management cycle, from the satellite tasking and image acquisition until the delivery of vector data and ready-to-print maps required by the user.

The main mapping products provided by the service are pre- and post-event maps related to a specific disaster event.

The pre-event or reference map provides geographic information of the territory before the event and are based on satellite images and geo-spatial data acquired prior to the disaster event.

The post-event maps assess the geographical extent of the disaster event in the delineation maps or evaluate the intensity of the damage resulting from the event in the grading maps; both products are based on satellite images acquired in the aftermath of the disaster event. The post-event maps are provided by the service within 12 hours after image reception and quality acceptance while 9 hours are required for the reference maps. The service also delivers within 2 hours after the image reception, a First Estimate Product (FEP) that roughly identifies the most affected locations.

The main critical aspect of the Copernicus Emergency Management Service Rapid Mapping (CEMS-RM) is the time required to provide all the relevant information in the aftermath of an event. Since the satellite images are the main source of information, it's clear that early-tasking of satellite acquisition could improve the time delivery of the products. Image request cut-off time is specific for each satellite platform and missing this cut-off times could mean to lose the opportunity of image data acquisition, leading to a consequent delay in the delivery of the products.

In the specific case, the output of LEXIS models could be used to trigger satellite tasking over a possible affected area afterwards an alert and before the activation of the service by an authorized user, moving up the delivery of the products.

Indeed, the output of the models, crossed with some exposure datasets, such as population estimation or number of buildings, could be used to prioritize some Areas of Interest (AOIs) to be submitted to the satellite data provider.

We expect that the integration of LEXIS output within CEMS-RM could also affect the FEP that, as mentioned before, provides a very fast yet rough assessment of the most affected areas. The output of LEXIS models could represent an effective way to generate a FEP because some outputs derived from models and integrated with available exposure data, provide information that can be exploited to quickly generate first damage estimation, before the availability of any post-event images.

The scope of the following activation description is to build up a dataset to be used as a benchmark, to be compared with a scenario simulation in which LEXIS outputs are available and exploited in the framework of a CEMS activation.

5.1 CEMS RAPID MAPPING ACTIVATION IN NEPAL

An earthquake with a magnitude of 7.9 M, 10 km depth and epicentre located between the capital Kathmandu and the city of Pokhara occurred on 25 April 2015 at 06:11 UTC (at 11:56 Nepal standard time), destroying houses, historical buildings and villages.

The CEMS was triggered on the same day at 10.24 UTC by ECHO ERCC to produce reference and damage assessment maps (grading maps) with the aim of mapping main affected areas and damaged level of buildings and industrial facilities in the hot spot areas at 1:10.000 scale (see Figure 11).

22 hours after the activation of the Rapid Mapping Service, first Reference maps covering Kathmandu, Bidur, Bharatpur and Pokhara have been generated. Damage assessment maps based on satellite image acquired on 27 April 2015 (bad weather conditions on 26 April 2015) and on 29 April 2015 for Pokhara (bad weather conditions in the previous 3 days) have been consequently produced.

Following:

- On 27 April 2015 it was required to produce reference and damage assessment map for a new AOI at 1:10.000 scale (Hetauda). The grading map was delivered on 01 May 2015.
- On 28 April 2015 there was a new request for 6 AOIs: the focus is to analyse the conditions of isolated villages in the north of Kathmandu at a lower map scale (map scale 1:50.000) in order to delineate affected settlements, landslide areas, presence of road block and gathering of people. 6 reference maps and 6 delineation maps were delivered respectively on 29 April 2015 and 02 May 2015.
- On 29 April 2015 there was the request for a new map over Gorkha, in a radius of approximately 5 km around the hospital with the aim to support and inform teams deployed in that area. The grading map at 1:10.000 scale was delivered on 30 April 2015.
- On 01 May 2015, 4 new additional AOIs at different map scale were requested to map damages in small towns and villages and to map landslides which may block rivers and road accessibility. 2 grading maps at 1:10.000 scale and 2 delineation maps at 1:50.000 scale were delivered on 03 May 2015.

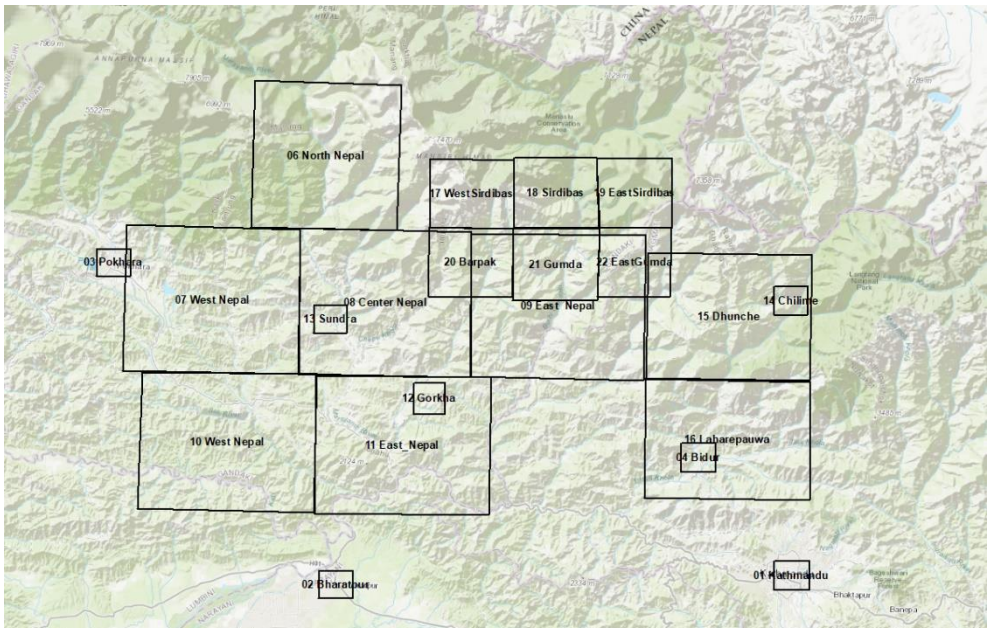

Figure 11 Location of areas mapped by CEMS-RM.
Details and related products of the activation are available on the website under activation code [EMSR125]: Earthquake in Nepal (https://emergency.copernicus.eu/mapping/list-of-components/EMSR125)

- On 09 May 2015 the service was activated for 6 new AOIs to produce mapping of roads and trek paths with related blockages and landslides affecting these at 1:25.000 scale. The delineation maps were delivered on 10 May 2015.

5.2 CEMS RAPID MAPPING IN CHILE

An earthquake of magnitude 8.3 M, at depth of 25 km, occurred off the coast of Choapa province in Coquimbo Region, central Chile, on 16 September at 22.54 UTC. The epicentre of the earthquake was just off the coast of Coquimbo province and the shaking lasted for approximately 3 minutes. A wave of up to 4.8 m was measured in the port of Coquimbo. A Red tsunami alert was issued for the entire country.

The Pacific Tsunami Warning Centre forecast tsunami waves in a larger area, including the coast of central, South and North America, as well in south and north western Pacific Ocean, from the French Polynesia to Japan.

CEMS-RM and the International Charter Space and Major Disasters (Charter) were triggered about 9 hours after the event. In their first request the Emergency Response Coordination Centre (ERCC), acting as authorized user of CEMS-RM, required maps in the area of Coquimbo city and surrounding, on the basis of media reports and knowledge from the ground. The service produced 9 reference and 9 damage assessment maps (grading maps) with the aim of mapping main affected areas and damaged level of buildings and industrial facilities at 1.5000 scale.

The results of first damage assessment maps were delivered 25 hours after the Earthquake and 16 hours after the user request (see Figure 12).

On 17 September at 17.26 UTC, the service was activated for additional 6 areas along the coast, on the basis of GDACS Tsunami Model alert. The delivered grading maps, with map scale above 1:10.000, revealed only minor damage or no damage (see Figure 13).

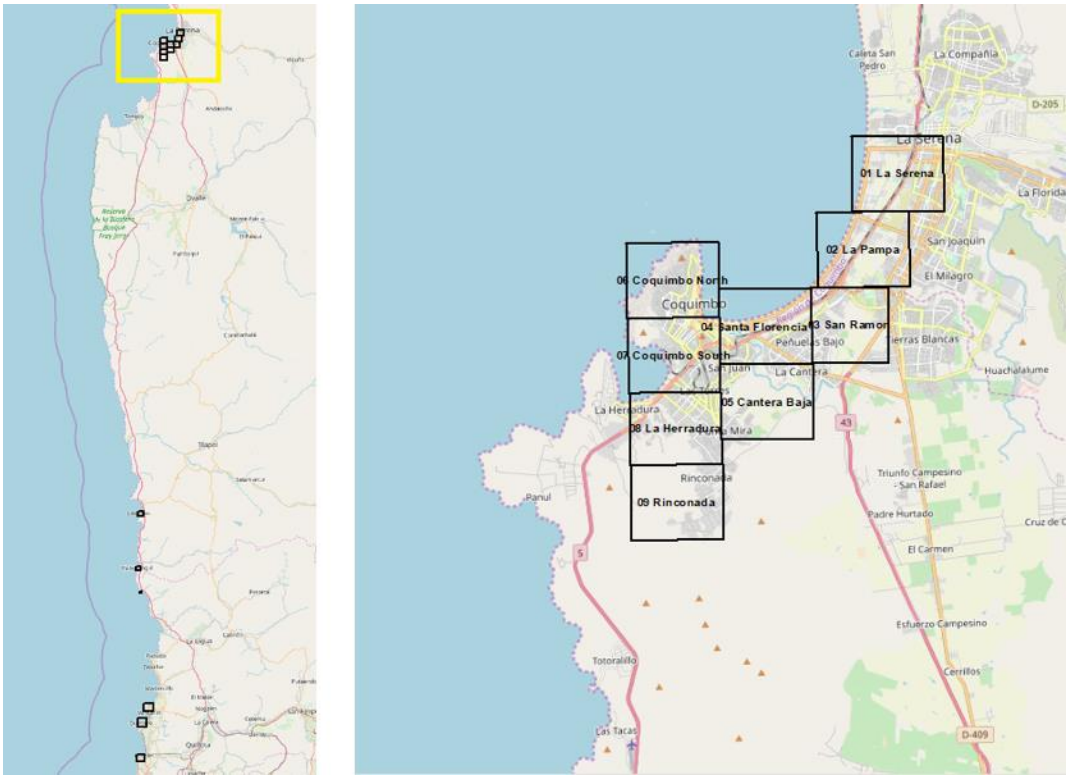

Figure 12 Location of the first 9 areas near Coquimbo mapped by CEMS-RM.
Details and related products of the activation are available on the website under activation code [EMSR137]: Earthquake in Chile (https://emergency.copernicus.eu/mapping/list-of-components/EMSR137)

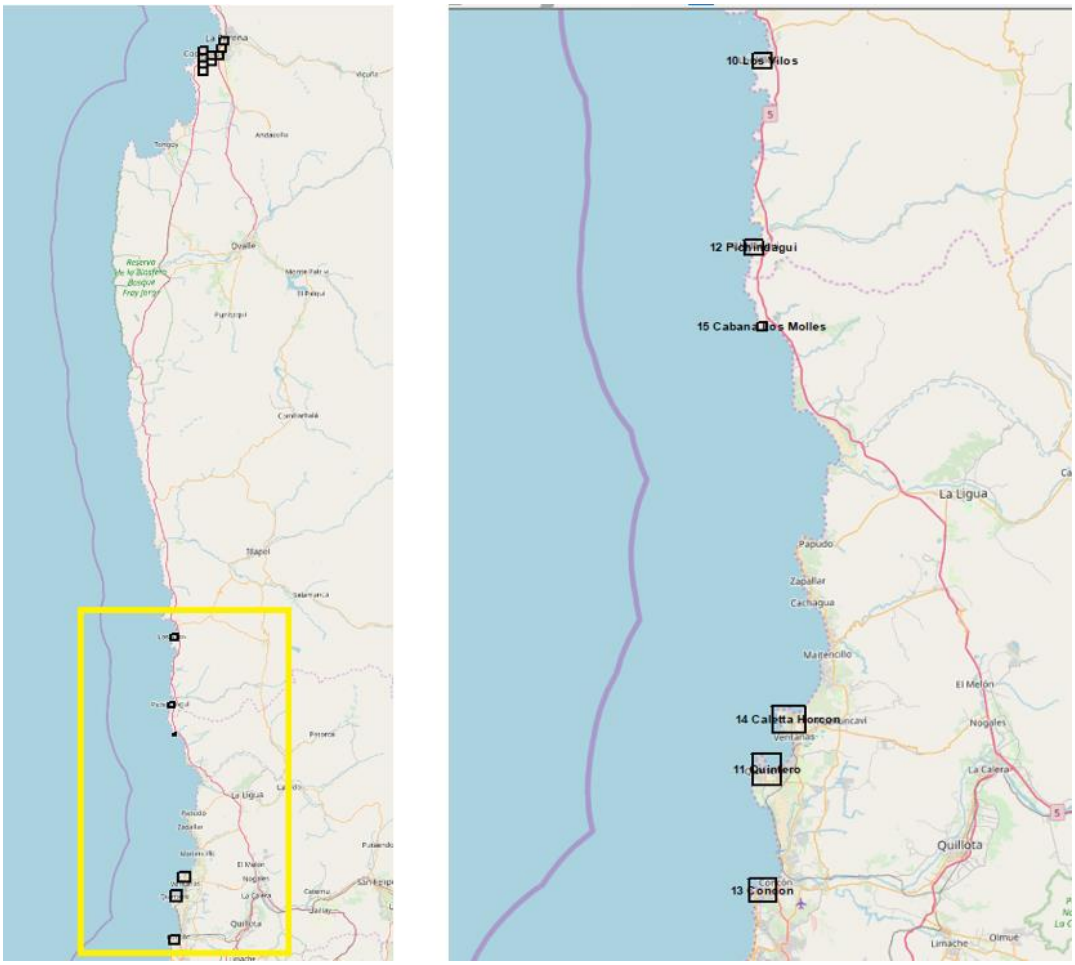

Figure 13 Location of the 6 areas along the coast mapped by CEMS-RM.
Details and related products of the activation are available on the website under activation code [EMSR137]: Earthquake in Chile (https://emergency.copernicus.eu/mapping/list-of-components/EMSR137)

5.3 OBSERVATION

Both CEMS RM Activations demonstrated the importance of integrating alert systems and exposure dataset in a satellite emergency workflow.

In detail in Nepal activation:

- The access to information on possibly affected areas derived from an alert system integrated with exposure information, could have made possible the delivery of a FEP without waiting for the acquisition of post disaster images, acquisition delayed for few days due to bad weather forecast;
- A building exposure dataset could have allowed a faster identification of the main affected areas and a better delineation of the areas of interest. Many areas, specifically, revealed only minor damage or no damage.

In Chile activation:

- The availability of a well accurate tsunami model in an alert system and updated exposure information could have allowed a better identification of the main affected areas along the coast. In the activation, for example, some grading maps were delivered on non-affected areas;

- An automatic prioritization process, able to identify the main areas of interest in the shortest time, could maybe avoid delay in the post-event image acquisition and false positive in the identification of affected areas.

The preparation of these 2 benchmark datasets will allow to quantify and analyse the potential benefits of a solution as the one proposed within LEXIS, assuming it was available at the time of the 2 events.

6 LINKS WITH OTHER WORKPACKAGES

The earthquake and tsunami large scale pilot main links are with the technical workpackages, especially the orchestration and data layers. The first one is about the pilot specific requirements and solutions, concluding to a feature support request in the shape of deadlines in the orchestration, significantly easing the implementation of the pilot, and, on the data layer, an understanding of the transparency and data movements allowing HPC jobs and Cloud jobs to cooperate and be linked via data products in the workflow. Another exchange happened at the infrastructure level, where the use of burst buffer technology was discussed to accelerate the OpenBuildingMap database update and querying.

This pilot is also linked with the WP7 weather pilot, where the objective is to combine weather and disaster situation during or in the period after an event. An interesting collaboration point would be a unification of the area of interest concept, that could be used to trigger additional processing and the merging of weather information onto the inundation and loss assessment results.

7 CONCLUSION

This document presents the evolution and work undertaken on the components of the earthquake and tsunami large scale pilot. It charts how the workflow itself has evolved, its scheduling being detailed, and how the components have been improving during the first part of the Task 6.2: the model of computation, OpenBuildingMap, the ShakeMap generation, and the TsunAWI tsunami simulation code, as well as the directions we are focusing on for the remainder of that task.

REFERENCES

[1] T. Goubier, A. Ajmar, C. D'Amico, P. Dubrulle, S. Grita, S. Louise, J. Martinovič, T. Martinovič, N. Rakowsky, P. Savio, D. Schorlemmer, A. Scionti and O. Terzo, "Earthquake and Tsunami workflow leveraging the modern HPC/Cloud environment in the LEXIS project," in *Proceedings of 22nd International Conference on Network-based Information Systems (NBiS)*, 2019.

[2] V. Silva, H. Crowley, M. Pagani, D. Monelli and R. Pinho, "Development of the OpenQuake engine, the Global Earthquake Model's open-source software for seismic risk assessment," *Natural Hazards,* vol. 3, no. 72, pp. 1409-1427, 2014.

[3] P. Dubrulle, C. Gaston and N. Kosmatov, "A Data Flow Model with Frequency Arithmetic," in *Proceedings of International Conference on Fundamental Approaches to Software Engineering (FASE)*, 2019.

[4] A. Musa, O. Watanabe, H. Matsuoka and H. Hokari, "Real-time tsunami inundation forecast system for tsunami disaster prevention and mitigation," *Journal of Supercomputing,* vol. 74, p. 3093–3113, 2018.

[5] A. Androsov, S. Harig, J. Behrens and J. Schröter, "Tsunami Modelling on Unstructured Grids: Verification and Validation," in *Proceedings of the International Conference on Tsunami Warning (ICTW)*, Bali, Indonesia, 2018.

[6] G. Gibb, R. Nash, N. Brown and B. Prodan, "The Technologies Required for Fusing HPC and Real-Time Data to Support Urgent Computing," in *IEEE/ACM HPC for Urgent Decision Making (UrgentHPC)*, Denver, CO, USA, 2019.

[7] B. Posey and et al., "On-Demand Urgent High Performance Computing Utilizing the Google Cloud Platform," in *IEEE/ACM HPC for Urgent Decision Making (UrgentHPC)*, Denver, CO, USA, 2019, 2019.

[8] F. Løvholt, S. Lorito, J. Macias, M. Volpe and J. Sel, "Urgent Tsunami Computing," in *IEEE/ACM HPC for Urgent Decision Making (UrgentHPC)*, Denver, CO, USA, 2019.

[9] J. Mandel and et al., "An Interactive Data-Driven HPC System for Forecasting Weather, Wildland Fire, and Smoke," in *IEEE/ACM HPC for Urgent Decision Making (UrgentHPC)*, Denver, CO, USA, 2019.

[10] C. Yepes-Estrada, V. Silva, J. Valcárcel, A. B. Acevedo, N. Tarque, M. A. Hube and et al., "Modeling the Residential Building Inventory in South America for Seismic Risk Assessment," *Earthquake Spectra,* vol. 1, no. 33, p. 299–322, 2017.

[11] S. Brzev, C. Scawthorn, A. W. Charleson and L. Allen, "GEM Building Taxonomy (Version 2.0), GEM Technical Report," 2013-2.

[12] LEXIS Deliverable, *D6.1 Baseline scenarios and requirements.*

[13] P. Dubrulle, C. Gaston and N. Kosmatov, "Dynamic Reconfigurations in Frequency Constrained Data Flow. IFM 2019:," in *Proceedings of Integrated Formal Methods, 15th International Conference (IFM)*, Bergen, Norway, 2019.

[14] R. A. Finkel and J. L. Bentley, "Quad trees a data structure for retrieval on composite keys," *Acta Informatica,* vol. 4, pp. 1-9, 1974.

[15] R. Ramakrishnan and J. D. Ullman, "A survey of deductive database systems," *The Journal of Logic Programming,* vol. 23, no. 2, pp. 125-149.

[16] N. Rakowsky and A. Fuchs, “Efficient Local Resorting Techniques with Space Filling Curves Applied to the Tsunami Simulation Model TsunAWI,” in *IMUM - The 10th International Workshop on Multiscale (Un-)structured Mesh Numerical Modelling for coastal, shelf and global ocean dynamics*, Bremerhaven, Germany, 2011.