Spatial Data Engineering for Digital Agriculture

Rob Knapen

Propositions

1. Semantic interoperability is the Achilles' heel of digital agriculture. (this thesis)
2. Effective connections between spatial data, models, systems, and researchers are a requirement for impact with digital agriculture. (this thesis)
3. Digital agriculture is amplifying inequality faster than it reduces hunger.
4. The peer review process is an essential slowing down of scientific progress.
5. Placebos work because the brain prefers meaning over molecules.
6. The whole universe is one big quantum simulation.
7. Scuba diving is a perfect way to get off-the-grid.

Propositions belonging to the thesis, entitled

Spatial Data Engineering for Digital Agriculture

Rob Knapen
Wageningen, 10 October 2025

Spatial Data Engineering for Digital Agriculture

Rob Knapen

Thesis committee

Promotor:
Prof. Dr I. Athanasiadis
Professor of Artificial Intelligence
Wageningen University & Research

Co-promotor:
Dr *ir.* S. Janssen
Group Leader, Earth Observation and Environmental Informatics
Wageningen University & Research

Other members:
Prof. Dr *ir.* P.W.G. Groot Koerkamp, Wageningen University & Research
Prof. Dr S. Reis, Deutsches Zentrum für Luft- und Raumfahrt, Bonn, Germany
Dr Z. Zhao, University of Amsterdam, Informatics Institute
Dr J.A.M. Koch, Wageningen University & Research

This research was conducted under the auspices of the C.T. de Wit Graduate School of Production Ecology & Resource Conservation (PE&RC)

Spatial Data Engineering for Digital Agriculture

Rob Knapen

Thesis
submitted in fulfilment of the requirements for the degree of doctor
at Wageningen University
by the authority of the Rector Magnificus
Prof. Dr C. Kroeze,
in the presence of the
Thesis Committee appointed by the Academic Board
to be defended in public
on 10 October 2025
at 3:30 p.m. in the Omnia Auditorium.

Rob Knapen
Spatial Data Engineering for Digital Agriculture
184 pages.

PhD thesis, Wageningen University, Wageningen, the Netherlands (2025)
With references, with summary in English

DOI https://doi.org/10.18174/681755

Summary

Digital agriculture marks a transformative phase in food production, propelled by the convergence of digital technologies such as remote sensing, the Internet of Things, big data analytics, artificial Intelligence, and autonomous machinery. These technologies promise to improve productivity, sustainability, and resilience in agricultural systems. However, to harness their full potential, robust geospatial data engineering practices are essential. This thesis explores how spatial data engineering can be designed and implemented to support the ongoing digitalisation of agriculture, offering a conceptual and empirical foundation for the development of Agriculture 5.0.

The central thesis is that spatial data engineering functions not only as a technical enabler but also as an integrative and human-centric discipline. Four critical dimensions of connectivity are identified: connecting data, models, systems, and researchers. Each of these is investigated through dedicated empirical chapters and is accompanied by applied case studies and peer-reviewed research.

Chapter 2 focusses on connecting data and evaluates the challenges posed by the characteristics of big data: volume, velocity, variety, and veracity. The work identifies semantic heterogeneity and data veracity as especially pressing issues for agricultural applications. Although semantic technologies such as ontologies and Resource Description Framework (RDF) triples offer theoretical solutions, their real-world adoption remains limited due to complexity and lack of tooling. The chapter advocates for lightweight semantic approaches and FAIR data principles to improve data interoperability and trust.

Chapter 3 explores connecting models through the use of OpenMI, a standard for integrating models across different domains. This research demonstrates the operational feasibility of model linking and highlights the importance of semantic and spatio-temporal coherence. The role of OpenMI in facilitating data exchange among soil, weather, and crop models is highlighted, and the findings suggest the need for standards that are both technically robust and easy to adopt.

Chapter 4 addresses connecting systems by implementing a scalable simulation infrastructure using Apache Spark and the WISS-WOFOST crop model. The performance and efficiency of the system are evaluated in cloud and high-performance computing

configurations. The results show substantial speed improvements in simulation runtime and aggregation tasks, demonstrating the potential of distributed computing in agricultural forecasting. However, the complexity of such systems poses barriers to adoption by agronomists and agricultural data scientists.

Chapter 5 examines connecting researchers through the lens of Virtual Research Environments (VREs). The chapter evaluates VREs as platforms for collaborative, interdisciplinary digital agriculture. A mixed methods assessment indicates that VREs can improve productivity, reproducibility, and user satisfaction when properly integrated into research workflows. The chapter concludes that co-designed, user-informed digital tools are vital to foster trust and engagement across stakeholder groups.

The synthesis in Chapter 6 integrates the findings and articulates the societal and ethical implications of spatial data engineering. It calls for transparent, explainable, and human-centred design in digital agriculture to counter the risks of opaque, technocratic systems. The discussion juxtaposes technocratic (AI-optimised) and agroecological (low-tech, farmer-centric) futures, showing how spatial data engineering is relevant to both scenarios. The research positions spatial data engineering as a conduit between technology and practice, enabling knowledge flows and supporting inclusive innovation.

In conclusion, this thesis argues that geospatial data engineering is fundamental to realising the promise of digital agriculture. Its success requires interdisciplinary collaboration, stakeholder inclusion, and ethical oversight. By bridging the gaps between data, models, infrastructure, and people, spatial data engineering becomes a cornerstone of sustainable, scalable, and human-aligned agricultural innovation.

❄

Contents

Chapter 1

Introduction

"All around me were machines, busily at work, machines that threshed and winnowed grain... A picture of machines and no man to control or watch them! Machines that seemingly with full consciousness walked out into the fields to do their daily work. And even now there was no living being among them save myself... Had these machines in some incredible fashion been provided with brains?"

- The Hidden Colony, von Hanstein, 1935

1.1 Context

1.1.1 Transformations in agriculture

Approximately 12,000 years ago in the First Agricultural Revolution, the traditional hunter-gatherer lifestyle was replaced by permanent settlements and animal domestication. After a long time, this was followed by the second revolution that occurred from the 17th century onwards with the reorganisation of farmland, following the end of feudalism in Europe. The 3rd (green) revolution introduced chemical fertilisers, pesticides, and new high-yield crop breeds alongside heavy machinery in the 1950s and 1960s. The era in which we are now in is referred to as the fourth agricultural revolution, or the *digitization of agriculture*, by the manifestation of the Fourth Industrial Revolution (Schwab, 2017) within the agricultural sector. This *digital agriculture* increasingly uses various technologies such as sensors and robotics and aims to optimise agricultural practices using data-driven methods.

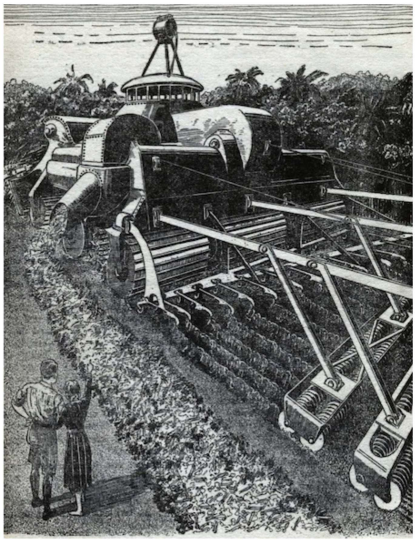


Figure 1.1: An autonomous farm robot. Illustration by Frank R. Paul depicting the mechanized farm scene in The Hidden Colony (von Hanstein, 1935), translated from Die Farm des Verschollenen (von Hanstein, 1924).

The earlier paradigm *precision agriculture* (Franzen and Mulla, 2016) (1980s - 1990s), which became possible, among others, by the development of the Global Position System (GPS) and Variable-Rate Application (VRA) farming machinery, paved the way for digital agriculture, incorporating and integrating new technologies and seeking data-driven agriculture solutions. Precision agriculture aims to maximise crop yields and farm profits

while reducing environmental costs by ensuring the optimal use of water, fertilisers, and phytosanitary products (Chlingaryan et al., 2018; Ruiz-Real et al., 2020; Schieffer and Dillon, 2015). Another variant of digital agriculture, called smart farming (Wolfert et al., 2017), goes one step further and incorporates informed decision-making based on data and context awareness (Sundmaeker et al., 2016). Smart farming is expected to start to bridge the gap between farming and Artificial Intelligence (AI) (Karunathilake et al., 2023), allowing the further development of advanced autonomous farm robots, such as those envisioned many years ago (Figure 1.1). Advanced integrated robotics and AI, combined with next-generation network technology and driven by Big Data, are expected to be crucial to Agriculture 5.0 (Fountas et al., 2024).

The transformation from traditional agriculture that relies highly on farmers' intuitions and experimental decision-making, with coarse site-specific farm management, to modern agriculture, particularly digital agriculture with precise field and in-field optimised farming activities, is considered crucial for addressing the many challenges faced by agriculture and our food systems. Such challenges result from global climate change, demand from an increasing population, geopolitical struggles, and better informed customers (Gebbers and Adamchuk, 2010; Moran et al., 2008; Slavin, 2016). Meanwhile, current systems and their extensive use of resources are also causing new problems such as decline in biodiversity and soil degradation (Stoate et al., 2009). Changes towards more resilient and sustainable food production and food supply chains are urgently needed. Increased digitisation in agriculture is commonly considered a necessity for a transition to "do more with less", that is, produce more (sufficiently nutritious) food while using fewer resources, to address the growing levels of food insecurity facing many countries (Food and Agriculture Organization of the United Nations (FAO) and International Fund for Agricultural Development (IFAD) and United Nations Children's Fund (UNICEF) and World Food Programme (WFP) and World Health Organization (WHO), 2022).

Benefits are expected by many, however, progressing traditional agriculture and agriculture science to become more data-driven has proven to be a challenging undertaking and a rather slow moving transition. Reviews of the literature on the topic carried out around 2017 (Kamilaris et al., 2017; Wolfert et al., 2017) found only a small number of publications and concluded that the adoption of big data processing and analytics was only in an "embryonic" stage at that time with many barriers to increased adoption still to be addressed. In particular socioethical and sociopolitical challenges, but also a call for science to strengthen the evidence base for digital agriculture and prompt the required new thinking (Ingram et al., 2022). Furthermore, Shepherd et al. (2020) mention that addressing the challenges should go hand in hand with solving the related technological issues, and agriculture science has to make a similar transition, including the potentially required organisational changes. Today, the technologies that are being researched and available for smart agriculture have improved significantly (Barbosa Júnior et al., 2024; Paudel et al., 2025). However, their adoption is still trailing, as is our understanding

of how to assess the barriers that cause it (Osrof et al., 2023; Talero-Sarmiento et al., 2022).

Although transition and adoption are slow, the possibilities of digital agriculture continue to evolve through rapid developments in technology and data sources, currently leading to the coexistence of different practices and paradigms (Figure 1.2). (i) *Model-centric digital agriculture* focusses on the optimisation of models (typically mechanistic models), using experimental data, and treating them as static. The experimental data results from controlled trials and research adhering to strict scientific protocols. (ii) *Data-centric digital agriculture* that focusses on rich and high quality data, mostly non-experimental but collected from real-world farming activities, processed by machine learning models and big data technologies. (iii) *AI-centric digital agriculture* that seeks to combine both paradigms, infusing existing physical and agricultural knowledge into data-driven machine learning to gain the best of both worlds (Roscher et al., 2023).

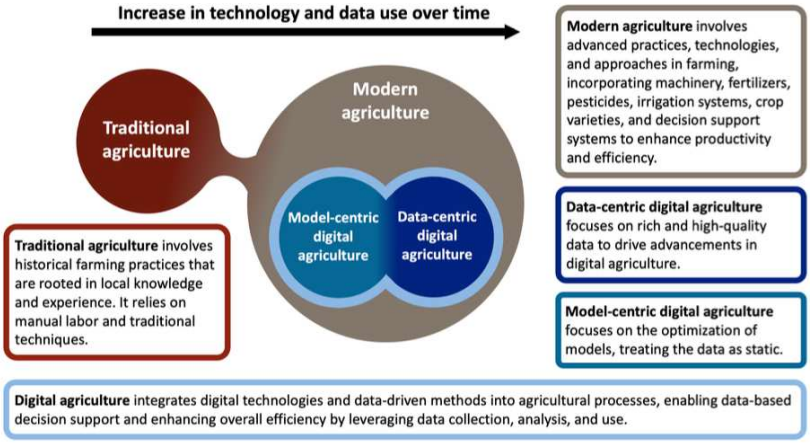


Figure 1.2: Transformations in Agriculture, illustration from Roscher et al. (2023)

1.1.2 Geospatial data engineering

One commonality between the various paradigms of digital agriculture is a high dependency on *spatially explicit data and models,* i.e. data that record locations or positional details of observations or phenomena and models that take them into account. Spatial data can describe the distribution of things in any kind of space, e.g. the human body, a room, or outer space. Geospatial—a concatenation of 'geography' and 'spatial'—indicates that the focus of the data is on features relating to planet Earth. Non-geospatial data, on the other hand, is data that is independent of geographic location. *The terms 'spatial' and 'geospatial' will be used liberally in this thesis.*

The fact that *location matters* makes spatial data special. Nearby events usually are more correlated than distant ones (Tobler's (1970) first law of geography, see Waters (2017)), requiring special techniques to handle such spatial autocorrelation problems. For example, in spatial data mining—which is the process of discovering non-trivial, interesting, and useful patterns in large spatial datasets—the most common of such spatial pattern families are co-locations, spatial hotspots, spatial outliers, and location predictions (Golmohammadi et al., 2020; Li et al., 2015). Such relationships among spatial attributes are usually *implicit*, while those among non-spatial attributes are *explicit*. That is, the fact that two spatial boundaries are typically neighbours is not explicitly described, whereas tractor X having fuel consumption Y will be directly represented in a data record.

Another distinguishing aspect is that while non-spatial data have the familiar types of nominal, ordinal, interval, and ratio, those of spatial data are *raster* and *vector*. Raster data are composed of grid cells or pixels and vector data of points, lines, and polygons. All are described using *coordinates*, which require *coordinate reference system* to relate them to a location on the (curved) Earth, and/or *transformations* to project them onto a flat surface such as a screen or a piece of paper. Such projections, unfortunately, are never ideal and always have to choose either not to distort angles and shapes (conformal map projections), not to distort areas (equal-area map projections), or to not distort line length (equidistant map projections) (Hargitai et al., 2017).

Finally, working with geospatial data requires understanding and using specific file formats, databases, tools, and algorithms (such as R-tree indexing (Guttman, 1984)), which are surrounded by many standards and organisations involved in development and support. Historically, this has been the domain of *Geographic Information Systems* (GIS) and *Spatial Data Infrastructures* (SDI, Hu and Li 2017), providing shared and national or even global access to many essential spatially interoperable data sources and the computational algorithms for processing them. GIS professionals typically work with powerful (desktop) computers and large (removable) storage devices, access data via Internet services that may be part of SDIs and based on well-known open standards such as those managed by the Open Geospatial Consortium (OGC, http://www.ogc.org). And/or make use of cloud-based (web) GIS for computational and storage-demanding processing of remote sensing data, such as satellite imagery, for near real-time insights. The currently emerging integration of AI and GIS as the field of *Geospatial Artificial Intelligence* (GeoAI, Gao et al. 2023) will only further increase the demands and complexity of the domain, expanding it to the field of *spatial data science* (Figure 1.3).

Spatial data science builds on GIS by integrating advanced statistical, computational, and machine learning techniques. Where GIS focusses on spatial data organisation and traditional geospatial analysis, spatial data science broadens the scope, using these foundational tools to uncover deeper insights, make predictions, and solve complex geographic

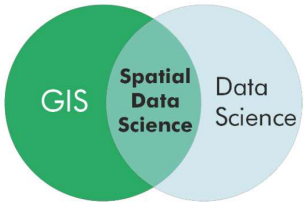


Figure 1.3: Spatial data science is the intersection of GIS and Data Science, and focuses on the unique characteristics of spatial data, moving beyond looking at where things happen to understand why they happen there. SDS treats location, distance and spatial interactions as core aspects of the data using specialized methods and software to analyse, visualise, and apply learnings to spatial use cases (Li et al., 2023a).

problems. An overview of some of the differences between GIS and spatial data science is given in Table 1.1.

Table 1.1: An overview of the differences between GIS and Spatial Data Science.

Aspect	GIS	Spatial Data Science
Focus	Spatial analysis and mapping	Advanced analytics and prediction
Methods	Traditional geospatial tools (buffering, overlays)	Machine learning, spatial statistics
Tools	GIS software (ESRI ArcGIS, QGIS)	Python, R, Scala, big data platforms (Apache Spark, Apache Flink, etc.)
Scope	Primarily spatial data	Integration of spatial with non-spatial data

As spatial data science itself evolves, a clear distinction from *spatial data engineering* is also required to address the growing complexity of data workflows. Spatial data engineering emphasises the infrastructure and processes required to efficiently manage, store, and process spatial data. This includes tasks such as designing geo-databases, ensuring data quality, optimising spatial data pipelines, and integrating diverse data sources. Spatial data engineering is a specialisation of more generic data engineering, which is itself a branch of computer science that deals with managing the creation, storage, maintenance, use, and dissemination of data. It uses programming languages such as Python, SQL, R, and Scala that aid in the manipulation of big data, and in many, if not all, cases it is known to be the most time-consuming aspect of data science (Hosseinzadeh et al., 2023). By separating the roles, spatial data engineers provide robust and scalable frameworks on which spatial data scientists can rely to analyse and extract value from increasingly large and complex spatial datasets.

> "Data Engineering is the development, implementation, and maintenance of systems and processes that take in raw data and produce high-quality, consistent information that supports downstream use cases, such as analysis and machine learning. A data engineer manages the data engineering lifecycle,

> beginning with getting data from source systems and ending with serving data for use cases."
>
> - *Fundamentals of Data Engineering, Reis and Housley (2022)*

Spatial data science and spatial data engineering play an important role in modern digital agriculture, where location-specific insights are crucial to optimise agricultural practices. Without adequate spatial data engineering, several challenges arise, including:

i. Inaccurate farm-level insights: Poorly managed spatial data can lead to inaccuracies in mapping farm parcel boundaries, soil types, or crop distributions, directly affecting decision making.

ii. Inefficient integration of diverse data sources: Digital agriculture relies on the integration of satellite imagery, IoT sensor data, weather models, and other data sets. Lack of data engineering makes it difficult to standardise, merge, or process these diverse sources, reducing the effectiveness of spatial data science in generating actionable insights.

iii. Slow or inconsistent analysis: Large-scale agricultural data, such as satellite imagery or drone-based surveys, require robust processing pipelines to handle high-volume and high-velocity data. Inefficient systems slow down analysis, which can delay critical farming activities, such as pest management or irrigation scheduling.

iv. Challenges with real-time decision making: Modern digital agriculture increasingly depends on near-real-time data, such as live weather updates or sensor feedback. Without a strong spatial data engineering foundation, delivering timely insights becomes difficult, undermining the value of spatial data science in operational decisions.

v. Limited scalability and accessibility: As farm operations scale or expand across regions, spatial data systems must handle larger datasets and provide insight across geographies. Weak data engineering can limit the scalability of spatial data science solutions, particularly in supporting smallholder farmers in low- or middle-income countries (LMICs).

vi. Reduced predictive accuracy: Spatial models are heavily dependent on clean and high-quality datasets. Poorly engineered systems can introduce noise or bias into data, leading to inaccurate predictions for yield estimation, climate risk analysis, or crop health monitoring.

vii. Barriers to knowledge sharing and collaboration: In digital agriculture, shared platforms and collaboration between stakeholders – farmers, policymakers, agronomists and researchers – are vital. Inefficient spatial data systems hinder interoperability, create silos, and reduce the effectiveness of collective decision-making.

Neglecting spatial data engineering undermines the effectiveness of spatial data science, limiting its potential to provide actionable insights and impact decision-making processes for modern agriculture and related environmental sciences.

1.2 Research gaps

In geospatial data engineering, there are still several critical research gaps that hinder its full potential, in addition to its slow adoption. These gaps can be identified at the various levels that are involved in the processing of raw data into information, the interpretation of that information into knowledge, and the effective application of that knowledge to make better decisions. Specifically, the need for scientific and application improvements in heterogeneous data integration, the integration of spatially aware computational models, the use of scalable distributed computing in digital agriculture, and the facilitation of interdisciplinary collaboration and knowledge sharing will be highlighted.

At the *data level*, modern digital agriculture is based on a wide array of spatial data sets from satellites, drones, proximal sensing, and ground-based agricultural monitoring systems. And since agriculture is the largest interface between humans and our natural environment, it cannot be considered in isolation. Other kinds of data sets from various research disciplines have to be included as well, for example about soil, climate, and biodiversity. Together, these form large to big data collections that have to be handled by data engineering. Suitable technologies exist, but little is known about their applicability for data-driven agriculture.

Once raw data are pre-processed and structured into *information*, computational models (including mechanistic, machine learning, and hybrid approaches) are used to further process them, or features extracted from them, to generate insights, predictions, or classifications. These outputs are structured information as well, ready to be interpreted by humans or systems to derive knowledge. Often there are multiple models in play, e.g. for interdisciplinary research, integrated assessment studies, or when the interacting components of complex systems have been modelled separately. This introduces the problem of interoperability; existing models often function in isolation, making it difficult to integrate the outputs or to use the output of one model as input for the next one in a chain of models. In case of spatial-aware models, it can be even more tricky due to differences in handling the spatial aspects of the data, and differences in spatial and temporal scales, e.g. between fine-resolution (farm, field and intra-field) and coarse-resolution (regional or global) agricultural data.

Since modern agriculture is data-driven, it will depend on capable data storage and compute facilities to handle data pre-processing, information processing, and transformation into *knowledge*. This requires a robust infrastructure, such as compute clusters and the use of scalable computational frameworks. While common in other scientific domains, e.g.

bioinformatics and climate research, their use for digital agriculture is under-explored. However, existing infrastructures often struggle to handle the volume and complexity of agricultural geospatial data in particular. Furthermore, the lack of proper and shared infrastructure leads to siloed systems and platforms operating within closed ecosystems, making it difficult to standardise and collaborate on workflows for spatial data ingestion, processing and analysis, for digital agriculture.

Finally, effective application of *knowledge* to make *better decisions* in digital agriculture requires productive communication and collaboration between researchers, engineers, and agricultural stakeholders. Several barriers currently limit such interdisciplinary approaches, including: (i) fragmentation of research communities that leads to disconnected advancements; (ii) limited adoption of the principles of *Findable, Accessible, Interoperable, or Reusable (FAIR)* data (Wilkinson et al., 2016), restricting the sharing of knowledge; and (iii) lack of centralised openly accessible repositories for agricultural geospatial data sets and research findings, which limits innovation and application. Portals, science gateways, and virtual research environments are known technologies that support e-research; however, their adaptation and deployment within the digital agriculture sector are relatively under-explored.

1.3 Research objectives

This thesis investigates and enables spatial data engineering for use in digital agriculture by testing and incorporating technological developments in agricultural applications. Based on the mentioned research gaps, it identifies key challenges towards: (i) connecting data; (ii) connecting models; (iii) connecting systems; and (iv) connecting researchers. Addressing them is crucial to facilitate not only geospatial data engineering but also directly and indirectly improve geospatial data science for modern data-driven agriculture. The research framework for the thesis is given in Figure 1.4, and its content is structured as follows:

Connecting data is discussed in Chapter 2, based on the paper "Analysis of Big Data technologies for use in agro-environmental science", which looks at the characteristics of big data (Volume, Velocity, Variety, and Veracity), the multidisciplinary *data – information – knowledge – wisdom (DIKW)* pyramids that typically have to be dealt with in digital agriculture, and answers the research question of which aspect(s) of Big Data we best focus on from a perspective of spatial data engineering in agriculture.

The Chapter 3 on **connecting models** is based on the publication "Evaluating OpenMI as a model integration platform across disciplines". The Open Modelling Interface (OpenMI) is a standard accepted by the Open Geospatial Consortium (OGC) that enables the run-time exchange of spatial data between numerical models and modelling tools. This chapter examines the standard and the approach of model linking in a number of research

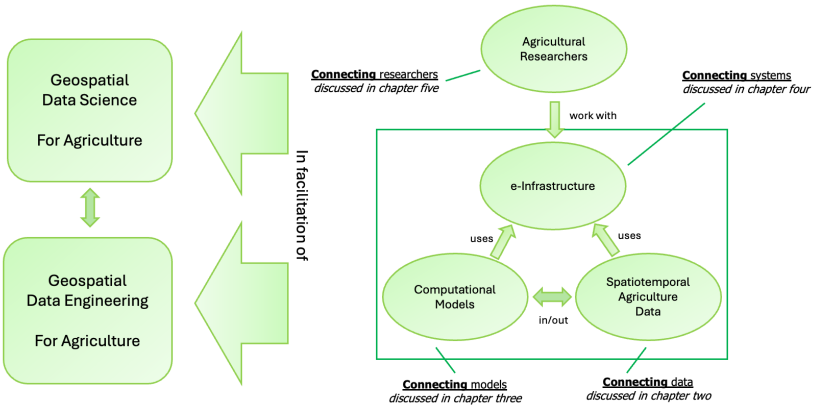


Figure 1.4: The research framework of this thesis

domains, driven by the research question if and how such a standard can facilitate spatial data engineering work.

Next, in Chapter 4, **connecting systems** is discussed based on the manuscript "Efficient and scalable crop growth simulations using standard big data and distributed computing technologies". It looks at the applicability of of-the-shelve available software and hardware for running a numerical crop growth simulation model at scale, i.e. for calculating millions of simulated crop yields on a compute cluster, compared to more traditional solutions by building bespoke systems. When successful, having such standard capabilities available would be beneficial to agricultural data engineering.

Finally, **connecting researchers**, is studied in Chapter 5 with the paper "Evaluating Virtual Research Environments in agri-climatic research". While the previous chapters address research questions related to geospatial data engineering, here the research question is related to Data Science and researchers in digital agriculture, particularly across domains, and whether and in which form Virtual Research Environments can be helpful.

The Synthesis, in Chapter 6, integrates the key findings of the research and highlights their collective contribution to the field. The work is situated within the broader context of modern digital agriculture, evaluates its relevance in light of recent advances in spatial data engineering, and discusses limitations. It also identifies remaining research gaps and outlines future directions to enhance geospatial data engineering for digital agriculture.

❄

Chapter 2

Connecting Data

This chapter is based on:

R. Lokers, M. Knapen, S. Janssen, Y. Van Randen, and J. Jansen (2016b). "Analysis of Big Data technologies for use in agro-environmental science". *Environmental Modelling & Software* 84, 494–504. DOI: https://doi.org/10.1016/j.envsoft.2016.07.017

2

Abstract

Recent developments like the movements of open access and open data and the unprecedented growth of data, which has come forward as Big Data, have shifted focus to methods to effectively handle such data for use in agro-environmental research. Big Data technologies, together with the increased use of cloud based and high performance computing, create new opportunities for data intensive science in the multi-disciplinary agro-environmental domain. A theoretical framework is presented to structure and analyse data-intensive cases and is applied to three case studies, together covering a broad range of technologies and aspects related to Big Data usage. The case studies indicate that most persistent issues in the area of data-intensive research evolve around capturing the huge heterogeneity of interdisciplinary data and around creating trust between data providers and data users. It is therefore recommended that efforts from the agro-environmental domain concentrate on the issues of variety and veracity.

2.1 Introduction

Societal challenges (e.g. food security, ecosystem restoration, climate change, resource use efficiency as captured in the Sustainable Development Goals (http://sustainabledevelopment.un.org/topics/sustainabledevelopmentgoals) and EU's societal challenges (http://ec.europa.eu/programmes/horizon2020/en/h2020-section/societal-challenges) require more and more complex approaches in terms of combining cross-sectoral and cross-discipline knowledge, information and data. For example, Steffen et al. (2015) introduce the concept of planetary boundaries to define a safe operating space for humans in the earth system, and thereby using data and models coming from many different domains and background. Such integrated scientific and societal perspectives require the combination of a multitude of data sources and the application of different analytical techniques.

Traditionally, science has operated along disciplinary lines in using and applying its data and analytical tools. Data management and curation was hardly an issue, with data being connected and analysed for separate applications and with researchers working with data les on their own computers and not actively publishing or sharing these. In roughly the period 1985–2005 there was a large focus on developing models for knowledge derivation from available data, see for example a review of farm models in Janssen and Van Ittersum (2007), crop models in Van Ittersum and Donatelli (2003), ecological models in Schmolke et al. (2010), land use models in Verburg et al. (2004). This period was followed in 2000–2012 by a period of building modelling frameworks as a method of combining more comprehensive analysis for decision making (e.g. Argent 2004; Knapen et al. 2013; Van Ittersum et al. 2008; Van Meijl et al. 2006), combined with many information technology and computational innovations to enable rapid analysis of large amounts of data within a single discipline (e.g. Villa et al. 2009). As a consequence, at this stage the capabilities within disciplines for data processing and analysis are well developed, just as the high-level linkage of models in abstract modelling frameworks, even if the methodological framework underlying such efforts is often lacking (Janssen et al., 2011). Looking at Wang's Levels of Conceptual Interoperability Model (Wang et al., 2009), in environmental modelling and simulation there has been substantial and useful progress at the lower levels of technical and semantic interoperability. To advance, besides further addressing these lower levels, also the still unexplored higher levels of semantic and conceptual interoperability have to be targeted.

Fortunately, in recent years a number of trends have emerged that could fundamentally change this status over the coming decade. First and for all, the mentioned trend of broadening policy and decision contexts research has challenged the science domain in general towards much more multi-disciplinary and integrative research, while the pace of decision making also puts pressure on the timeliness of research results. Second, the political attention has turned to open data as public good resource, as witnessed by open data initiatives (e.g. Global Open Data for Agriculture and Nutrition, www.godan.info), open

data conferences and open data portals (e.g. data.gov, data.gov.co.uk, data.overheid.nl, data.fao.org). This development was preceded by a movement to make scientific publications available as open access, which has led to specialised journals being set up and traditional journals offering the option to publish under open access licences. Third, the amount of data available for science has grown enormously in the past years, driven by technology developments such as open access repositories of remote sensing images, the advance of the mobile phone enabling crowd sourcing and citizen science and digital connectedness through social media and internet of things. Fourth, the computational resources have massively increased over the past decades, according to Moore's Law, with also a better availability and accessibility of storage and computational resources in the cloud such as Platform-as-a-Service (PaaS) and Model-as-a-Service (MaaS) technologies.

These developments of more (open) data and higher connectedness in principle offer opportunities to support larger, faster and more complex data-intensive processing and analysis across disciplines as required for supporting evidence-based decision making towards societal challenges. Against this background, recently Big Data has emerged and to some extent has been hyped as a new trend to provide unlimited capabilities in analysis of data, providing revolutionary new insights (boyd danah and Crawford, 2012; Manyika et al., 2011; McAfee and Brynjolfsson, 2012). Related to the agro-environmental domain, Vitolo et al. (2015) have investigated web technologies dealing with "Big Environmental Data", while Lokers et al. (2015) explore the use of semantic technologies to improve access to Big Data in agriculture and forestry science. For the purpose of this paper, Big Data is defined as: *a term encompassing the use of techniques to capture, process, analyse and visualise potentially large datasets in a reasonable timeframe (as defined in Networked European Software and Services Initiative (NESSI) 2012), while incorporating both structured and unstructured data and covering several disciplines and domains.* This definition primarily focusses on technology and on the technological support of some of the elementary data-intensive tasks in science. Use cases on data management in research (Lokers et al., 2014) show a variety of technological challenges associated for instance with environmental modelling, that range from metadata oriented information retrieval issues to heavily data-oriented problems related to Big Data mining and data integration. These challenges in particular concern the effective discovery of the appropriate data for a specific research task. In data-intensive research areas like agro-environmental modelling we have reached the point where automated procedures for selection, collection and indexing are becoming indispensable to effectively exploit this global network of data.

In this paper we examine and analyse use cases from three European projects as guidance to describe current possibilities and future challenges for deployment of Big Data techniques in the field of agro-environmental research, facilitating decision support at the level of societal challenges. For that purpose, a theoretical framework is proposed that allows positioning of Big Data challenges and techniques in the context of interdisciplinary science and the policy-science interface. This framework is then applied to analyse three scientific

cases in the agro-environmental domain and to reflect on the current state of play of the application of Big Data technologies in the domain. Based on the analysis of the cases along the theoretical framework, overall observations are made on technology readiness and suggestions are provided for further developments.

2.2 Analysis

2.2.1 Theoretical framework

It is useful to start from a theoretical framework framing the complexity of challenges and demystifying the hype of Big Data. Such a theoretical framework needs to be tailored to the context of the agro-environmental domain. To achieve this, Big Data, its characteristics and ways of processing should be connected to the context of evidence based decision making and to the specifics of data-intensive challenges in the agro-environmental domain.

To frame the way (big) data is used in decision making we introduce a knowledge management model, extending a broadly used and recognised concept which has been elaborated on in numerous publications in different forms and under different names and to which we will refer here as the data-information-knowledge-wisdom or DIKW hierarchy (Rowley, 2007).

The model (see Figure 2.1) is used to contextualise data, information, knowledge, and sometimes wisdom, with respect to one another and to identify and describe the processes involved in the transformation of an entity at a lower level in the hierarchy (e.g. data) to an entity at a higher level in the hierarchy (e.g. information). The idea is that decision makers need "wisdom" for taking evidence based decisions. Such wisdom can be developed by combining available knowledge with less tangible assets like interests, values, preferences, ethics, etc. The knowledge base they use is essentially derived from data. Data can in this respect be considered the raw material to produce information through the addition of meaning. Information is again enriched, creating knowledge by using and combining decision and policy contextual applications like for instance integrated models, impact assessments or decision support systems.

Agro-environmental research use cases usually concern dynamic systems with complex interactions between living organisms or perishable products (e.g. plants, animals, humans, agricultural products) and their environment. Describing such systems requires complex and usually detailed information regarding status and behaviour of its entities and their environmental conditions. It can include its actual status, but also historical or predicted future conditions. Because of the spatial dynamics and temporal variability of living systems, data regarding the temporal and spatial behaviour of entities and local conditions are essential. Moreover, understanding these interactions requires the observation, analysis and integration of knowledge of subsystems of very different nature, for example

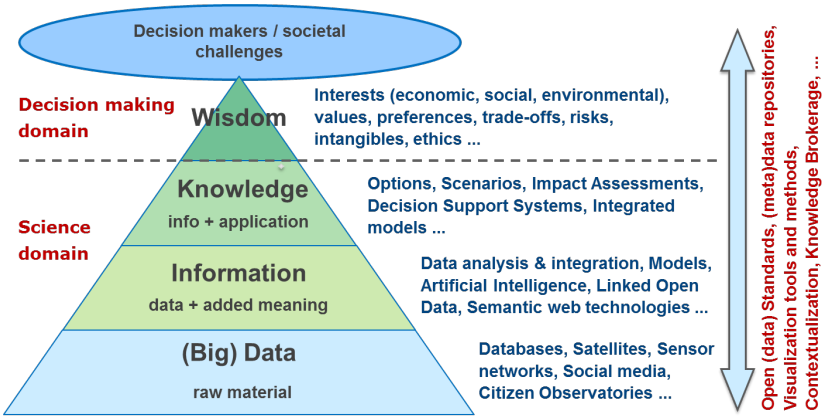


Figure 2.1: DIKW hierarchy, from Big Data to decision making for societal challenges

biological, climate, soil and water subsystems. The complexity of describing, analysing and understanding such systems and the magnitude and heterogeneity of the data involved can be easily understood.

The complexity of handling Big Data is highly associated with its typical characteristics, often described as the "3 V's" of Big Data, i.e. *Volume*, *Variety* and *Velocity* (Laney, 2001).

Volume refers to the unprecedented amounts of data becoming available through new technologies supporting massive generation or collection of data and efficient means of storage. Relevant examples for the agro-environmental domain include climate data (especially climate projections) and remote sensing data. Terabyte to Petabyte size volumes are easily reached when attempting to capture – for example – natural variability on detailed spatial and temporal scales.

Velocity refers to the speed at which new data is becoming available, e.g. through real-time data streams, but also refers to the usually high requirements regarding processing time to make the data and its value-add derivatives available for end users. In the agro-environmental domain, real-time data generated by sensor networks or citizen science networks are good examples of such streams, while monitoring and early warning systems commonly require near-real-time processing of such data streams in order to provide timely information to decision makers.

Variety concerns the ever increasing heterogeneity of data relevant to decision-making. Firstly, this is caused by the continuous evolvement of available streams and formats, e.g. from social media and mobile applications. Moreover, information from an increasing range of disciplines is needed, in particular in the agro-environmental domain. This is

due to the many subsystems of very different nature, the tremendous width of current societal challenges to be addressed, and the resulting complexity of associated decision contexts. Because individual disciplines tend to have a background of working in silos and using their own tailored data formats and vocabularies, these attempts to integrate data or information from different domains face a multitude of technical and semantic challenges.

In addition to the three V's mentioned, additional characteristics of Big Data have been identified. *Veracity*, often mentioned as being "the fourth V" (http://www.ibmbigdatahub.com/infographic/four-vs-big-data), seems to be the most relevant one when we specifically consider the agro-environmental domain. Veracity, which addresses among others the integrity and accuracy of data and data sources, is highly associated with trust and with having confidence that the quality of data is sufficient to serve as evidence base for critical decision making. Researchers will have to leave the safe environment of familiar data silos in peer networks, while at the same time the growing size and complexity of the data ecosystems grows beyond the capacities of a human being to judge the quality of all associated data sources. Consequently, frameworks and working procedures that ensure integrity of data and its derived products and trustworthy indicators for integrity become indispensable.

Figure 2.2 shows how these Big Data characteristics are linked to the DIKW layers when we also consider that in most agro-environmental cases multiple disciplines are involved, with different content regarding data, information and knowledge and different perspectives on policy and decision making.

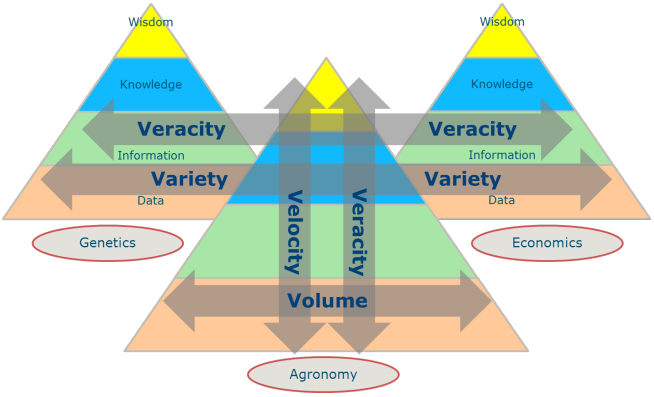


Figure 2.2: Multidisciplinary Big Data pool and characteristics

In the context of Big Data, the DIKW hierarchy also conceptualises the process of turning the enormous mass of data, which as a raw material has little or no significance to end users, into compact, structured and contextualised, manageable "chunks" that are applicable

in a specific decision making context. End users will implicitly presume that these have been synthesised using the most appropriate sources from the Big Data pool, interpreted, and processed according to their decision context, using the most reliable and timely information available. Evidently, such presumptions pose an enormous challenge to the whole community of ICT-experts, data scientists and domain experts that are involved in handling the various steps in this process. The broad scope, both vertically over different ICT, data science and knowledge management expertise areas and horizontally, covering the multi-disciplinary of present-day decision contexts, requires a highly cooperative approach and the establishment of harmonised concerted processes, organised through a combined top-down and bottom-up approach.

To explore the possibilities to meet the challenges described above, in the next Section three data-intensive use cases from the agro-environmental domain will be described and analysed with regard to their position in the theoretical framework and the associated Big Data characteristics. Table 2.1 summarises the linkage of the cases with the Big Data characteristics and the DIKW model described above.

Table 2.1: Analysed use cases characteristics

Use case	Volume	Velocity	Variety	Veracity	D	I	K	W
Semantic driven discovery		X	X	X	X		X	X
Data driven discovery	X		X	X	X	X		
Big Data querying	X	X	X		X	X		

2.3 Case: semantic driven discovery

2.3.1 Problem statement

This use case addresses the harmonised provision of scattered and heterogeneous data for impact assessment to decision makers and researchers. An impact assessment study typically requires assessing the potential economic, social, and environmental effects of alternative policy options through a number of scientific computer models that can span various science domains. Each of the models requires sets of trustworthy input data. For example, agricultural impact assessment studies could use scientific models such as: APES–a cropping system model (Donatelli et al., 2010); FSSIM–a bio-economic farm model (Louhichi et al., 2010); CAPRI–an agricultural sector model (Britz et al., 2009); and GTAP–a computable general equilibrium model for global markets (Hertel and Tsigas, 1997). Input data required would include, amongst others, crop parameter data, data on local soil types, historical and simulated future weather data (on local, regional, and global scale)). Many of such datasets are available, either locally in organization's repositories, or on the Internet as open data. Due to the expanding use of sensors and satellites that can

measure e.g. crop, soil and meteorological data at increasingly finer temporal and spatial resolutions, not only the amount of available datasets is growing, but also their sizes. This makes finding the usable pieces of data one of the key challenges of Big Data.

In an approach to address this discoverability challenge, the LIAISE project developed the LIAISE Toolkit (http://www.liaise-kit.eu). LIAISE–Linking Impact Assessment Instruments to Sustainability Expertise–was established in 2009 to improve the application of Impact Assessment (IA) by both the research and the policy making communities. The Toolkit facilitates the categorisation and discoverability of metadata for different types of knowledge resources related to IA, for example datasets, scientific models, frameworks, practical examples and domain experts. Submitted information is categorised into topics by key experts before it is published and made accessible through the Toolkit website. Initially this website supported directory based discovery with a faceted search mechanism. Following technological developments near the end of the project it was explored how the search capabilities could be improved by the use of semantic technology. In particular this included investigating whether and how recent Natural Language Processing (NLP) and Machine Learning (ML) techniques could be used to automatically derive required metadata from unstructured text sources and relate it to a defined LIAISE overall ontology. It was assumed that using these techniques could lead to a system which does not only rely on manual provision of metadata by experts, but which can also get its content from automated discovery of relevant metadata, or enriching existing sparse metadata from auxiliary documentation such as reports, published papers, or websites. It would support finding the relevant small pieces of data (at the DIKW Information layer), as well as make the system more "intelligent", operating at the Knowledge and Wisdom layers, linking available heterogeneous knowledge sources from multiple disciplines to specific contexts of decision and policy making. Furthermore, the case is strongly connected to the variety characteristic, establishing semantic links between the knowledge and decision-making layers and by exposing new opportunities for innovative re-using and combining tools in new domains. Through its foreseen approach of automated linkage, it also touches the aspect of improving velocity, while at the same attempting to retain veracity or trust in the generated knowledge.

2.3.2 Methodology and implementation

For its practical implementation, the case aimed at extending the existing LIAISE Toolkit with (i) a way to use the LIAISE ontology for linking existing external datasets to all already available knowledge resources in the Toolkit, and (ii) the use of a similar pathway to relate typical impact assessment questions to relevant knowledge resources in the Toolkit, providing a semantic search mechanism. Figure 2.3 illustrates the foreseen steps, including: (1) selecting and processing of auxiliary documentation of simulations models and datasets using NLP techniques, (2) mapping of the data to the de ned LIAISE ontology, and (3) storing it. For retrieval through a stand-alone web interface (6) or from the Toolkit website,

questions posed in natural language will be processed (4), related to the stored information and used to find (5) matching search results (i.e. relevant knowledge resources).

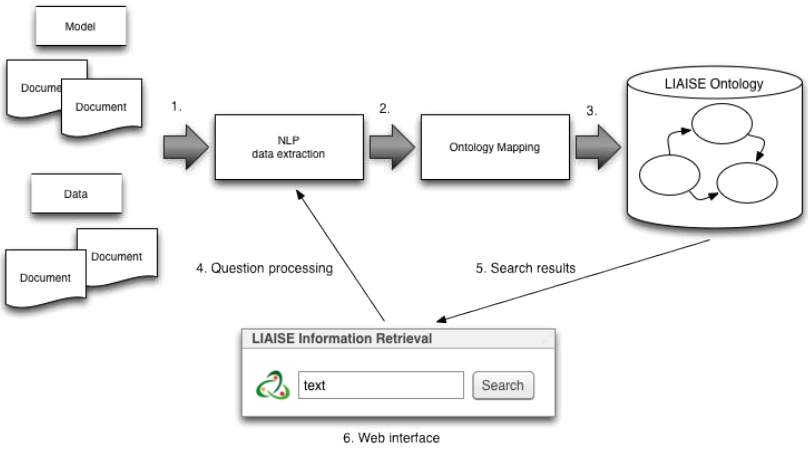


Figure 2.3: LIAISE semantic driven information retrieval concept

As a proof-of-concept the semantic linkage exercise was developed around the datasets provided online by the European Environmental Agency (EEA, http://www.eea.europa.eu/-data-and-maps). The metadata of these EEA datasets contains references to a list of topics relevant for EEA resources (see http://www.eea.europa.eu/themes). LIAISE, on the other side, uses a taxonomy of impact areas for tagging included knowledge items.

For reasons of performance and quality an automated procedure periodically retrieves metadata of available EEA datasets from their semantic web SPARQL endpoint. It then attempts to find and add relevant LIAISE taxonomy-based impact area tags to the metadata, thus establishing links that allow the LIAISE web portal to mention the datasets at appropriate places, for example, as potentially suitable input to a simulation model. To create the links, the automated procedure needs to perform some kind of semantic comparison based on available metadata and/or data. Several approaches for such a semantic comparison were explored and tested.

The first approach was based on the exploitation of Machine Learning and Natural Language Processing techniques, enabling computers to derive meaning from human or natural language input (Ng and Zelle, 1997). It foresaw the building of a corpus for the automatic determination of relevant terms from the metadata available through LIAISE knowledge resources to subsequently analyse and tag the external metadata with a Machine Learning algorithm. Unfortunately, at the time of development the Toolkit was just started to get filled by the experts and a corpus of adequate size could not yet be constructed within the project time boundaries. Existing resources such as the online,

publicly available dictionary WORDNET (Miller, 1995) do not contain the very domain-specific knowledge required for e.g. semantic parsing, and a sense-tagged corpus needs to be added to improve automated semantic interpretation. Finding sufficient material for building training data for machine learning was an additional, yet related problem. Consequently, initial ambitions for this case had to be scaled down.

A second, less elaborate approach based on textual matching techniques was subsequently explored. Using the OpenNLP tools (http://opennlp.apache.org), each textual description of a LIAISE taxonomy term for an impact area (e.g. "Environmental Impacts - The environmental consequences of firms and consumers - Sustainable production and consumption") was syntactically analysed to find all nouns in it, and compared to nouns found in any text field (title, description, topics, etc.) of each resource from the EEA ontology referring to a dataset. The more nouns matched, the more relevant the resource was considered, and the higher it was ranked in the search results. This simple approach proved to be relatively successful due to the fact that both EEA and LIAISE work in the environmental science domain and thus already use a kind of shared vocabulary. Their words and terms in most cases mean the same things. Yet, LIAISE topics and subtopics purposefully have broad and non-restrictive titles so that experts can always find one or more topics their knowledge resources fit in without having to define new topics. While this keeps the taxonomy stable, it makes it harder to use for machine processing. Hence precision and recall of the text matching turned out to be too low to make it an acceptable approach, and the search results contained too much noise over signal to make it acceptable to the users.

Therefore, the final implemented approach was an expert-driven linkage process. This method uses a mapping table in which the expert manually links the LIAISE impact areas (or taxonomy terms) to EEA thematic topics. It does not provide an explicit indication of the quality of the match, which is implicitly associated with the expert and their level of expertise. Because this mapping requires manual input by experts for each data provider to be added to the system and for changes in the taxonomies, it is more time consuming and less dynamic, but it does provide expert-based quality of the links, creating trust for the web portal users, thus addressing the veracity aspect.

2.3.3 Results

From the three explored approaches to link (EEA) datasets with the knowledge base available in the LIAISE Toolkit, the semi-manual method where experts manually link Impact Assessment terms to terms related to the datasets was implemented. The two automatic processes that were examined proved to be too costly or ineffective in this particular case. For the Machine Learning and NLP-based approach, the main barrier was that building a working corpus for this purpose from available LIAISE resources was not

possible, due to the lack of sufficient material available at the time of development, and the unforeseen amount of time that it would take.

Using textual matching, it appeared that in all cases the process resulted in considerable amounts of erroneous matches (low precision and recall values) producing undesired and unusable results. This could be at least partially assigned to the relative simplicity of the non-semantic textual matching techniques used in the process. Moreover, matching also failed because the process operated on extracted high level generic terms (water, air, pollution) instead of more specific compound terms (also known as n-grams) like "surface water" and "air pollution". Retrospectively it can be concluded that trust, or veracity, also plays a relevant role. Even if conditions are met to successfully implement automated procedures for tagging, it will remain hard at the moment for these technologies to gain the level of trust that the scientific user community tends to exhibit if experts perform the job manually.

2.4 Case: data driven discovery

2.4.1 Problem statement

The Trees4Future project (www.trees4future.eu) is an integrative European Research Infrastructure project that aims at integration and further development and improvement of major forest genetics and forestry Research Infrastructures. One of the objectives is to make forestry scientific data discoverable and accessible for a broad audience of modellers and decision makers in and outside the forestry research community. Like in many other scientific domains, forestry researchers traditionally rely on their own peers and scientific networks when collecting the data required for their work. Only recently the forestry research community has started to harmonise and share their data, especially in the area of genetics. However, a lot of relevant data is still stored in silos, sometimes even in local or private repositories. Moreover, datasets often are not documented with appropriate metadata. In many cases, researchers do not see the benefits of documenting data, or data is consciously kept private for example because associated research results are still to be published or because of fear for misuse. In general there is often no incentive, nor sense of urgency to actively share data other than through (trusted) networks and personal contacts. This corresponds to observations in literature, suggesting that apart from the technology challenges, many disciplines also still lack the institutional and cultural frameworks required for efficient data sharing, together leading to a "scandalous shortfall" in the sharing of data by researchers (Nature Editorial Board, 2009). Thus, valuable research data is hard to find without knowing the right people, and only partially available for the whole community of interest. Consequently it still remains hard to acquire the specific targeted data for interdisciplinary work. This lack of discoverability is an even more

pressing issue for "newcomers", for scientists from associated domains that require forestry data for their work or for decision makers looking for evidence-based information.

One of the research communities in the Trees4Future project are forestry modellers. Their work on present-day societal challenges (e.g. related to bio-economy, climate change) requires interdisciplinary approaches, like integrated modelling. As an example, assessing climate change impacts and exploring climate adaptation strategies requires coupling of models that describe various sub-domains and cover different spatial and temporal resolutions. In Trees4Future, such integrated assessments required the linkage of the ForGEM model (Kramer et al., 2013), the EFISCEN model (Nabuurs et al., 2000) and the Tosia model (Lindner et al., 2010). While the ForGEM model assesses genetic adaptive responses on the individual tree and population level, the EFISCEN model projects forest resource development on a regional and European scale, and the Tosia model analyses environmental, economic, and social impacts of changes in forestry-wood production chains. Heterogeneous data, varying from detailed genetic data, phenotypic traits and high resolution climate and soil data, to statistical data on species distribution, forest management practices and market information are required to address such integrated modelling exercises. Given the current disconnectedness and lack of context, it is quite complex and time consuming to discover and get access to these data. The Trees4Future project aims at improving this situation by developing technical solutions to facilitate the documentation, publication and discoverability of forestry data by setting up a forestry data infrastructure. Moreover, through this infrastructure it aims at demonstrating benefits and fostering broader uptake of data sharing and documentation practices.

From the perspective of the theoretical framework, this case is strongly connected to veracity, addressing issues of trust and quality. This obviously works in two directions. On the data owner side, there needs to be trust that data is sufficiently documented and will not be misinterpreted or misused. Data consumers, on the other side, should have trust in the reliability and correctness and completeness of associated metadata. The case also addresses variety, through the requirement to provide integrated access to sources coming from a range of relevant subdomains and the related need to provide semantic linkages over the associated (meta)data. This case mainly concerns the lower levels of the DIKW hierarchy (the data and information level) and the need to make the available data and information part of the multidisciplinary Big Data pool, adding the required context on the dataset level to make data potentially usable for the broader community.

2.4.2 Methodology and implementation

To improve access to data required for and generated by forestry research, a data search and discovery service on top of a federated metadata repository was developed. Main objectives were firstly that the system had to be able to provide both already documented and accessible datasets and up till now inaccessible datasets, thus connecting not only

organisations that have already organized and standardised their processes but also the smaller organisations and individuals that are not equipped with the required infrastructure. Secondly, end users were to be provided with a facility to easily search and discover available forestry data, once datasets have been documented and published through the developed mechanisms. This search function was considered to be the necessary "proof of the pudding", required to convince end users to use the system, but also to convince data owners of the benefits of documenting and publishing their data.

To achieve improved discoverability, the following components were developed to support the data publication process depicted in Figure 2.4:

i. A concise metadata schema, based on the widely supported and extensible Dublin Core standard and extended with additional metadata elements to support forestry specific metadata;

ii. A public metadata registry, composed of an online metadata editor and an underlying repository, which publishes its metadata records through the OAI-PMH protocol, providing a standardized and harvestable metadata endpoint;

iii. A forestry ontology that allows the conceptualization of datasets and its interlinkage with commonly used external ontologies (e.g. AGROVOC (http://www.fao.org/-agrovoc/), a genetic traits ontology);

iv. A metadata harvesting, triplification and annotation mechanism that supports harvesting metadata following the developed forestry metadata schema as well as standardized metadata schema's like INSPIRE and ISO; decomposes the metadata into ontology concepts using among others natural language processing (NLP) techniques and stores these in an RDF (Resource Description Framework) database; and links the derived dataset concepts to the concepts of the available external ontologies;

v. A semantic search mechanism and search interface, allowing users to transparently search the registered datasets, using the power of semantics in the underlying RDF store.

2.4.3 Results

The developed infrastructure clearly has increased the discoverability of forestry research data and improved its availability for a broader audience. It covers the variety of data required for integrated forestry modelling cases, like the described climate adaptation use case and others. This is, first of all, because it provides federated access to the currently scattered and sometimes inaccessible wealth of forestry research data. The developed data infrastructure already publishes metadata of more than 300 data sets from major European data repositories, and offers the option for small organisations and individuals to publish their (meta)data through a managed access point. Moreover, it offers opportunities to

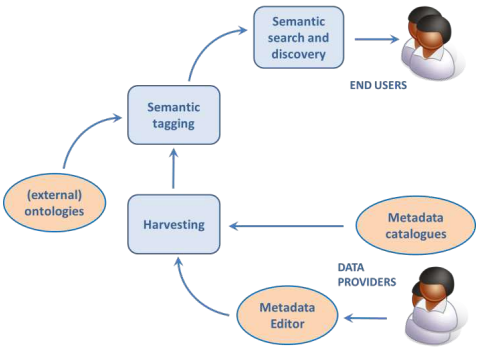


Figure 2.4: Publication process workflow developed in Trees4Future

publish reference datasets for integrated modelling, providing less experienced modellers with entry points to build their experiments.

The developed infrastructure has been tested with a set of queries to evaluate its added value through the use of semantics. Typical examples include the linkage of synonyms (e.g. rainfall results in datasets tagged with the concept rain) and broader and narrower terms (e.g. precipitation results in datasets tagged with rain, snow, hail). These tests, and the first impressions of its use in practice, show that even with relatively simple knowledge technology additions, well documented data can be made accessible in a better way, making it easier to discover data through better structuring, indexing and search capabilities. The addition of semantic capabilities and its ability to directly search topics, concepts and associations linked to a vast number of sources to a metadata repository increases the discoverability of datasets, because it reveals otherwise unknown linkages between the common vocabularies of different users and the actual metadata concepts. This is accomplished by (1) returning results that are semantically related to the provided search terms and (2) revealing related terms to the user when composing their search conditions (e.g. by a semantically driven autocomplete function). An observed additional benefit is that providing the scientific community with improved discovery mechanisms increases awareness that data documentation is important, and contributes to insights in how data could be best documented in order to provide added value to end users.

On the other hand, we also conclude that in forestry and related domains the currently available metadata is scarce and often of low quality, which complicates the linkage of metadata concepts with (external) ontology concepts. A second observation is that currently available metadata standards provide insufficient possibilities to (automatically) select datasets that fit researchers needs, e.g. in technical domains like modelling and simulation. In general metadata schemas lack the structure and depth required to structurally capture the complexity of scientific datasets. Commonly used and essential fields, like, for example,

lineage, do not provide the structure required to address the complex production processes of data. Moreover, the lack of depth prevents that the structure and contents of the data itself (for example its attributes and datatypes) can be addressed in a structured manner. In the use case, this issue was tackled by combining and linking isolated fragments of a broad coverage vocabulary (AGROVOC) with specific and detailed subdomain specific semantics. Obviously, this is a very customised and elaborate approach and not a viable generic solution.

2.5 Case: Big Data querying

2.5.1 Problem statement

Research in the agro-environmental domain has to deal with large and very diverse datasets, both in content, structure, and storage format. Because of the current move towards open access and open data, an increasing amount of data is brought out of their information silos and made accessible as part of what is called the Linked (Open) Data (LOD) cloud, resulting in an extensive network of distributed heterogeneous data sources. Unfortunately, access to this network to date is neither easy nor transparent, and current centrally-managed or even distributed data repositories are not able to meet the data science challenges ahead, starting with adequate Big Data querying facilities. The EU FP7 research project SemaGrow examined solutions to provide more effective and transparent ways to access distributed data. It aimed at developing algorithms and infrastructure for the efficient querying of large-scale federations of independently-managed data sources, i.e., the nodes of the Linked Data cloud. To address the differences in storage formats, it builds upon the already established and frequently used principles behind the Semantic Web, namely RDF and the SPARQL query language. These standards enable the sharing and reusing of data across applications and scientific community boundaries, and allow the interconnecting of data in the LOD cloud.

SemaGrow specifically focussed on the agriculture domain and its use cases through a series of data pilots, exploring the specific data challenges of this domain. These challenges typically include discovery, merging and integration of large and very diverse spatio-temporal datasets. One of the use cases explored in SemaGrow is regional agro-climatic modelling in the frame of climate adaptation. Climate parameters required for regional modelling are usually stored in large multidimensional les, often with global or trans-regional spatial coverage and long term temporal coverage. Modellers tend to duplicate large amounts of data for their modelling experiments, which are then locally processed to the required extent and scales. Besides the general issue of resource efficiency, such ways of working can pose significant barriers, specifically in regions where networking, storage and computing resources are limited. SemaGrow has examined ways to allow the thematic, spatial and temporal querying and merging of large distributed datasets,

returning relatively light and integrated datasets. As an example, this would allow an agricultural modeller in Ghana with limited networking and storage resources to acquire a merged subset of temperature, precipitation and soil parameters for a specific region in the country.

With regards to the theoretical framework, the case primarily focussed on the volume and variety characteristics of Big Data, exploring ways to allow scientists to efficiently access large, distributed data sources in a federated manner and to download and merge manageable subsets of different nature. Although velocity was not the primary focus, it is relevant to note that the case elaborates on automating data integration problems that generally are very labour and time intensive through the involvement of experts of different disciplines. While developed solutions might technically be considered as non-performant, they could still result in dramatic improvement of efficiency in the face of timely provision of information required for decision making. It concerns mainly the data and information layer of the DIKW hierarchy, attempting to efficiently bridge the gap between these levels by automatically processing and harmonising sources from the Big Data pool to a level that offers better opportunities for connecting data with the tools (e.g. models, data analysis) operated on the information level. Consequently, it not only potentially reduces the efforts and resources required to produce information from raw data, but also touches some of the integration challenges associated with interdisciplinary science.

2.5.2 Methodology and implementation

To be able to demonstrate SemaGrow Big Data querying capabilities in the frame of real-world applications, and to be able to compare its characteristics to a reference situation, as one of the pilots the Trees4Future Clearinghouse system described in the previous case was adapted to work using SemaGrow technologies. For that purpose, the Trees4Future back-end was replaced with the infrastructure developed by SemaGrow, the so-called SemaGrow Stack, and a set of distributed RDF databases containing triplified data and metadata. As a result, the demonstrator application also extends the reference application by offering the option to perform semantic queries on metadata but also on the underlying data.

The SemaGrow Stack (http://github.com/semagrow/semagrow) is a "federated SPARQL query processor" that can efficiently query a set of distributed heterogeneous data nodes. It includes a query planner that uses metadata about the nodes of the federation to optimise the query execution. This metadata follows the Sevod vocabulary (http://www.w3.org/-2015/03/sevod), also developed in the project, that extends the VoID vocabulary with statistical information akin to database histograms. The Stack uses the reactive software paradigm to properly handle unresponsive or slow data nodes in the federation. As such, the SemaGrow Stack provides a unifying endpoint that allows transparent querying

of the underlying triple stores without having to know their (possibly) heterogeneous schemas.

Triple stores were set up, holding triplified agro-environmental data, selected from the ISI-MIP and AgMIP data harmonisation initiatives. ISI-MIP, The Inter-Sectorial Impact Model Inter-comparison Project, is a community-driven modelling effort bringing together impact models across sectors and scales to create consistent and comprehensive projections of the impacts of different levels of global warming. Input and output data from ISI-MIP is made available as NetCDF les using the Climate and Forecast conventions for its metadata. The Agricultural Model Inter-comparison and Improvement Project, AgMIP, is a major international effort, linking the climate, crop, and economic modelling communities with cutting-edge information technology, to produce improved crop and economic models and the next generation of climate impact projections for the agricultural sector. AgMIP provides data in JSON format using the ICASA Variable List for its metadata. Both data collections are harmonised, but are quite different in nature, e.g. global gridded time-series data of simulation model projections, versus single point location based time- series of field management and weather station observed data. These features make them well suited to evaluate how the SemaGrow Stack handles the heterogeneity related aspects. Data from these sources has been "triplified" into triple stores, so they can be queried using SPARQL. Besides, the different vocabularies used (CF Conventions for ISI-MIP data and ICASA for AgMIP data) have been aligned through the use of the AGROVOC thesaurus. The amount of data sets that have been triplified for the demonstrator is limited. It concerns around 10 global coverage, long-term ISI-MIP data sets and a few dozen of AgMIP datasets. However, especially due to the volume of the ISI-MIP datasets, the total size was at the Tera triples level, allowing to also explore the volume related aspects and the associated behaviour of the SemaGrow infrastructure.

Lastly a spatio-temporal triple store (Strabon, http://strabon.di.uoa.gr) has been added to the federated nodes so that spatial queries, e.g. point-in-polygon, can be resolved. To connect the web application front-end of the demonstrator with the SemaGrow Stack instance, a small additional layer of middleware software was needed. It translates URL requests including parameter values into the proper SPARQL queries for the Stack, and vice versa preprocesses the raw query results into a response the demonstrator can handle. Furthermore, it is able to create valid NetCDF files from the RDF Data Cube format used internally, to better serve end-users needs.

2.5.3 Results

So far, the described demonstrator application has been tested by a limited group of end-users. The demonstrator gets positive remarks for the functionality it offers, but people expect better performance both for metadata searches (less than 10s expected, versus 5s–30s measured) and for data downloads (less than 30 min expected, versus several

minutes to several days measured, depending on the size of the selected data), as well as access to much more data. Both can possibly be met by massive upscaling of the infrastructure. Notably, in the performed expert enquiries, several experts have explicitly mentioned that, even with the measured response times, the demonstrated querying and data fusion facilities can be quite useful. It should be realised that, for example, in the formerly mentioned use case of agricultural modelling in Ghana, composing a dataset for modelling requires different processing steps. It usually requires consultation of, and cooperation with local and remote specialists, and consequently aggregated time investments and resulting lead times can be high. Thus, automated data-fusion queries, even when taking hours or days, could make the research process in such cases more efficient.

The SemaGrow project has also shown how time-consuming it remains to process data so that it is properly annotated with metadata, triplified, and aligned to make it part of the LOD. While tools for ontology matching and alignment were available through the project, these could not be used because reference vocabularies did not comply with the supported standards. Moreover, it appeared that a commonly used vocabulary like AGROVOC is not well suited to effectively annotate data sets on the level of detail required for the research problems examined. AGROVOC provides relatively rough concepts for specific variables and provides no specific unit taxonomy. However, fitting selections for specific modelling experiments would require more specific specifications to describe, for example, the parameter "mean daily temperature 2m above ground level" as well as its specific unit of measurement. Besides, in contrast to for example bibliographic data or text documents, the multi-dimensional data used in agro-environmental science still challenges state-of-the-art triple stores and current semantic web technology. Issues like different spatial projections, spatial and temporal scales, unit conversions, handling streaming data or simple data manipulations, could not be considered within the scope of the project, but were on the list of evaluation comments by the end-users. Consequently, providing transparent and unified access to these datasets is not yet trivial.

2.6 Conclusions and recommendations

Three use cases are described that have addressed different issues related to Big Data usage and technologies in the agro-environmental domain. These have also been put into the perspective of a theoretical framework to structure their complexity. In the analysed cases, a variety of issues were encountered spanning the whole range of Big Data characteristics (the 4 V's) and the layers of the DIKW hierarchy. Cases generally focused on discovering and combining heterogeneous datasets for modelling and decision making in interdisciplinary domains. While it is obvious that challenges regarding the volume and velocity aspects exist, and there are not yet clear solutions in all cases, the contours of future technical solutions are already visible, combining cloud-based storage and computing

with improved and better integrated infrastructural components. Research initiatives explore and develop innovative infrastructures and several commercial services are offered. More important for the agro-environmental domain is that steps are being taken to improve the handling of Big Data, including the aspect of dealing with the spatio-temporal data that are very common to the domain. Specific processing requirements of this type of data include spatial and temporal up- or down-scaling and handling a large variety of spatial reference systems. Such processing could be more effectively handled by an additional software layer, e.g. through a data centric design (http://research.ibm.com/articles/-datacentricdesign/), where much of the processing is moved to the places where the data is stored.

More persistent barriers in agro-environmental science, and probably also in other areas that require highly interdisciplinary knowledge for decision-making, lie in handling the variety and veracity aspects. Not surprisingly, these aspects are also crucial to link the different levels of the DIKW hierarchy, both vertically, allowing to work up raw data to knowledge fit for decision making, and horizontally, to meaningfully connect content from different disciplines that are currently often disconnected. In order to be able to meaningfully link heterogeneous sources coming from different disciplines and being generated for different purposes, improved semantic interoperability is needed. Possibly it also is needed to strive towards a higher level of the Conceptual Interoperability Model introduced by Wang et al., 2009. Based on the work done on the three presented cases, two approaches can be recognised, one top-down driven and the other one bottom-up.

The top-down approach would include defining and agreeing upon a top-level ontology for the agro-environmental domain, and all subdomains relating their specific ontologies to this top-level ontology by harmonising or alignment efforts. Many vocabularies and ontologies exist in the agro-environmental domain, developed with different purposes and covering different subdomains. They vary from broad coverage and relatively global (e.g. AGROVOC) to very specific coverage and detailed. The scope of agro-environmental scientific challenges usually requires dealing with different vocabularies that are typically not interoperable and sometimes even conflicting, which in practice makes it very hard to align semantics in such a way that the result remains meaningful and is fit for a specific purpose. Therefore, besides the elements of coverage and granularity, serious barriers are the fact that different standards are used, and that alignment is very labour intensive and requires interdisciplinary expertise. The analysed cases particularly show that standardised and broadly accepted semantics to describe datasets on the level of its attributes are generally lacking, and that available ontologies and vocabularies cannot easily be applied. Solutions like combining (fragments of) different semantic sources or manual linkage of vocabularies for very specific purposes can work for the specific case, but are obviously not sustainable. Yet, these, often small, semantic differences between simultaneously existing ontologies competing for adherents may simply continue to exist as part of our academic and political freedoms. Still it would be worthwhile to at least work towards common,

linked ontologies instead of ending up with, exaggerating, one ontology per disciplinary data silo.

A more bottom-up oriented approach revolves around the use of semantic interpretation technologies such as Natural Language Processing and Machine Learning algorithms. With more data, including e.g. text documents and web pages, and metadata becoming available, it will become impossible for humans to properly relate the data to ontologies, next to discussions about which ontologies to use. It certainly seems more practical when computers can tag data on an ad-hoc, case-by-case basis, based on some level of understanding of the meaning of the data. The WORDNET dictionary already provides a good starting point, but needs to be extended with domain specific knowledge, like discussed in the use case on semantic driven discovery. Building up such a corpus covering the agro-environmental domain is a time consuming activity, but it would be highly reusable in many future applications. Currently, the use of semantic technologies is not very well developed in the agro-environmental domain. Consequently, consistency of produced results is still varying, making its introduction and acceptance ("veracity") a challenge.

The sketched approaches are of course not mutually exclusive, and might meet somewhere in the middle. A relevant initiative that demonstrates a possible way forward is CYNERGI http://earthcube.org/group/cinergi). This initiative tries to combine bottom-up (e.g. enhancers using semantic techniques to improve metadata) and top-down (e.g. using a generic metadata schema) aspects to harmonise access to interdisciplinary datasets. More importantly, CINERGY recognises the shortcomings of available metadata, semantics, and available technologies, like machine learning and NLP. It explicitly includes human engagement as an indispensable factor in the process of making data fit for interdisciplinary science. Direct involvement of scientists to select relevant data sources, metadata elements and to validate generated metadata and query results is regarded a crucial element to serve cross-domain, fit-for-use data to scientists. This corresponds with the experiences and outcomes of the analysed cases.

Looking from the perspective of the analysed cases at the lower levels of the DIKW hierarchy, the provision of sufficient, high-quality metadata hinders the smooth access to and linkage of scientific data sources. To be able to work with the available metadata and to deal with its shortcomings, all observed cases were somehow confronted with the need to develop customised solutions. Applied solutions range from implementing manual ontology alignment as an alternative for metadata-driven automatic annotation to the improvement of awareness and provision of metadata creation and editing facilities. In general, despite the availability of standardised metadata schemas, well-documented scientific data is still scarce. This also appears to be a cultural issue, where scientific practice is often still based on working in silos and interchanging among trusted peers, and data management policy is not well developed. There is a clear link to the veracity aspect of Big Data here. Data

users need to trust the provided data, which is expressed most clearly by the quality of its metadata. On the other hand, data providers require trust that their data is used in a proper way, which again can be promoted by adequately documented datasets.

Promising and viable approaches for ICT driven mechanisms to improve interdisciplinary data-intensive research using technologies related to the Big Data domain have been identified, examined, and implemented in the analysed use cases. Although in most cases implementation was successful, we can also conclude that effectiveness is limited, due to the current state of data management and semantic coverage in the agro-environmental domain. Based on the analysed cases and the above-stated conclusions, we recommend that Big Data research, and especially the efforts to be delivered in this area from the agro-environmental domain (in contrast to the more technically oriented ICT research), focusses on variety and veracity challenges. This focus should lead to the improvement of conditions and development and application of methodologies and techniques required to efficiently provide access to and semantically interlink sources from different disciplines. Obviously, this does not only require technological advances, but also a disruptive change of culture and behaviour. To this end, the development and promotion of working demonstration cases in research environments has proven to be a valuable instrument to create awareness and catalyse such change.

Acknowledgements

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007–2013) under grant agreements No. 243826. (LIAISE), 318497 (SemaGrow) and 284181 (Trees4Future) and the Ministry of Economic Affairs of the Netherlands.

❄

Chapter 3

Connecting Models

This chapter is based on:

M. Knapen, S. Janssen, O. Roosenschoon, P. Verweij, W. De Winter, M. Uiterwijk, and J.-E. Wien (2013). "Evaluating OpenMI as a model integration platform across disciplines". *Environmental Modelling & Software* 39, 274–282. DOI: `https://doi.org/10.1016/j.envsoft.2012.06.011`

Abstract

For decision makers in the domains of agriculture and environment, for instance in government agencies, farmers, environmental NGOs and farmers' unions, it is beneficial to evaluate ex-post or to asses ex-ante the impacts of their choices. To research these interdisciplinary relationships, models developed by different scientific disciplines and often operating at different scales can be integrated into model chains that cover processes across disciplines. In order to assemble models into an operational model chain conceptual, semantic and technical levels of integration have to be taken into account.

The main focus of this paper is on technical integration to ensure repeatability and reproducibility of model chain runs and to optimise use of computer hardware for model simulations. Technical integration itself can be achieved by different approaches (i.e. manual, scripting, building or using a proprietary framework, using an open framework based on standards). From the many available modelling frameworks (e.g. OMS, TIME, KEPLER, FRAMES, MODCOM, OpenMI) the emphasis will be on OpenMI, the Open Modelling Interface and its use and usefulness as a readily available, generally accepted and open standards-based framework.

OpenMI is an open source software standard for dynamically linking models at runtime, which can potentially be used in many domains, but is currently mainly applied in the water and environmental domains. This paper describes and evaluates the use of OpenMI in several multi-disciplinary large projects that worked on integrated models. These projects operated in the disciplines of agriculture, land use, nitrogen cycling, forestry, hydrology, and economics.

To this end two workshops were organised to acquire feedback from both software developers and modellers that contributed to the aforementioned projects on the use of OpenMI. Perceived advantages and disadvantages of OpenMI differed between modellers and software engineers, although both identified the lack of standard functionality as a major disadvantage and the prescription of a way of working through OpenMI as a standard as a major advantage. In conclusion, OpenMI can be used as a standard for technical model integration across disciplines, and it is not limited to one particular discipline.

3.1 Introduction

For decision makers in the domains of agriculture and environment, for instance in government agencies, farmers, environmental NGOs and farmers' unions, it is beneficial to evaluate ex-post or to asses ex-ante the impacts of their choices. An ex-post evaluation occurs after such a choice has been made, while an ex-ante assessment tries to simulate the potential impacts of choices before these are made. In ex-post evaluation, data is likely to be available or can be collected on relevant variables in the period after the choice took effect. In contrast, an ex-ante assessment tries to shed some light onto the future and data is not available. Modelling and modelling tools can be helpful by providing a simplified representation of reality, simulating potential contrasting pathways into the future and improving the understanding of interdisciplinary cause-and-effect relationships. A model is defined as a deliberate simplification of reality that represents part of reality as a quantitative system.

Ex-ante assessments through models and modelling tools could provide valuable insights on potential choices for complex societal and environmental problems (e.g. climate change, achievement of the United Nations' Millennium Development Goals, as well as Millennium Ecosystem Assessment (Carpenter et al., 2009)). A prominent example of the use of an ex-ante assessment is the assessment of the likely impacts of climate change on the biophysical environment and society (IPCC) by the Intergovernmental Panel on Climate Change. An example on a lower spatial scale is the FARMSCAPE project (Carberry et al., 2002), in which farmers, advisory services and researchers jointly applied a simulation tool to assess the potential for alternative management strategies of cropping systems on Australian farms. Such ex-ante assessments need to involve multiple disciplines and cover multiple scales.

Mono-disciplinary models cover only a few processes from a single domain, be it economic, agricultural, or environmental and lack descriptions of some relevant processes. These models generally do not cover the relevant multiple scales to handle all assessment questions. Mono-disciplinary models can complement each other and thereby provide comprehensive and balanced assessments across scales. Therefore, it becomes necessary to integrate models from different disciplines, sometimes operating at different scales into model chains, covering processes across disciplines. In order to arrive at an operational model chain for applications in integrated assessment procedures, semantic, methodological and technical integration of models is required. Ideally, model integration leads to a model chain that can be trusted, is transparent, and can be used and understood by a community of researchers, not only the individual modeller.

Semantic integration means speaking a common language and achieving a shared understanding between all models and modellers working together (Hinkel, 2008; Jakobsen and McLaughlin, 2004; Scholten, 2008; Tress et al., 2007). The methodological integration

focusses on aligning different scientific methodologies and identifying required model improvements necessary for meaningful linkage. Methodological integration in a model chain requires that the data produced by one model are a meaningful input to another model, usually operating data at a different temporal and spatial scale (Liu et al., 2008). Finally, technical integration means automating data exchanges between models, making them jointly executable, without human intervention. Technical integration aims to ensure repeatability and reproducibility of model chain runs and to optimise use of computer hardware for model simulations (e.g. batch running, grid computing, cloud computing, high performance computing).

This paper focusses on technical integration and touches slightly on aspects of semantic integration while leaving methodological integration aside. Technical integration can be achieved using different approaches. First, the most straightforward method is soft linking: a manual and ad-hoc linking of models through output files of one model that are used as input files to another model after conversion, reformatting, and transformations in, for example, spreadsheet programmes through manual work. The EURURALIS project (http://www.eururalis.eu) is one of the many projects that apply this type of model integration. It is perfectly suitable when the integration does not have to be repeated often. Second, scripts (i.e. Python, Perl) can automate some of the manual steps in the conversion of output files to input files (for an example, see the description of the SENSOR project later on in this paper). Third, models are integrated in a proprietary method, where one model encapsulates the other model, making it possible to share input files and variables in memory, to implement feedback loops between models and to run models as one large monolithic model (SWAP - http://www.swap.alterra.nl/, MODFLOW – http://www.modflow.com/, WOFOST - http://www.wofost.wur.nl/). Fourth, instead of building a large monolithic model, a proprietary approach can be used to loosely link the models (IMAGE - http://themasites.pbl.nl/en/themasites/image/). In this approach, models become substitutable and the linking mechanism can be more transparent and better documented. The proprietary approach for linking the models has to be built and is specific for the models integrated or the organisation that develops and maintains it. Five, instead of a proprietary approach to link models, an available and generally accepted modelling framework can be used. Such a modelling framework can be based on open standards, making it easily accessible to a larger community. In such a model framework, the models are linked as independent components, exchanging data in a standardised way. Different modelling frameworks have been developed, each with their own setup and philosophy. The choice of one of the five approaches (i.e. manual, scripting, proprietary large model, proprietary framework, open framework based on standards) to technical integration depends on the institutional and project context and on the researchers involved.

This paper focusses on one particular modelling framework, OpenMI, used for linking models in an open framework based on standards. The paper describes the use of OpenMI

in several multidisciplinary large projects that work on integrated models. These projects operate in the disciplines of agriculture, land use, nitrogen cycling, forestry, hydrology, and economics. The objective of the paper is to investigate the strengths and weaknesses of integrated modelling according to open frameworks based on standards in general, and OpenMI in particular. It recommends a list of improvements for OpenMI specific and integrated modelling in general.

The next section Methods provides background on modelling frameworks for technical integration, OpenMI, the projects, and the method used in evaluating the use of OpenMI. In the subsequent Results section, the use of OpenMI across research projects and domains is described and contrasted, and strengths and weaknesses are extracted from the research projects. Finally, in the Discussion section, the main lessons in terms of using OpenMI are highlighted for future OpenMI development and for researchers working on integrated modelling.

3.2 Methods

3.2.1 Modelling frameworks

Following trends in software engineering and commercial software development and people trained in such modern methods entering the environmental science disciplines there have been several attempts to develop standardised approaches to modelling and model integration in the environmental problem domain, resulting in a number of modelling frameworks. A modelling framework brings together suites or libraries of modules with the intention to standardise features such as data manipulation and analysis, exchanges between models and data sets, structure an coding of modules, and visualisation of model outputs. The term *framework* is commonly used to refer only to the underlying classes and libraries, and *environments* to refer to systems that use these software frameworks to support module development, model construction, and execution (Argent et al., 2006). A further differentiation that can be made is that between the definition of a framework, the *specification* or interface, and its *implementation*. When not further specified, the term framework is used in its broadest sense in this paper.

Modelling frameworks have been conceived and developed for some time, with a varying range of characteristics (Argent, 2004; Jagers, 2010). For brevity's sake, further discussion will be limited to the following well-known frameworks with which the authors have at least some level of practical experience: the Open Modelling Interface (OpenMI - Moore and Tindall, 2005), a software standard for dynamically linking models at runtime, which can potentially be used in many domains, but is currently mainly applied in the water domain. TIME (Rahman et al., 2003) which is, like the OpenMI, a generic computational framework for building and executing models that may be applicable across domains. MODCOM (Hillyer et al., 2003) which is used for linking biophysical process-based models

in crop growth simulation. Moore et al., 2007 propose the Common Modelling Protocol which nests dynamic models in a hierarchy with a common interface on top and also focusses on dynamic and biophysical models. It is still at the core of the Agricultural Production Systems Simulator (APSIM). The Object Modelling System (OMS), a reusable domain-specific framework developed by the US Department of Agriculture (David et al., 2002). The operational modelling environment FRAMES (Framework for Risk Analysis of Multi-Media Environmental Systems), in use by the US Environmental Protection Agency (EPA) (Whelan et al., 1997), and Kepler, a general-purpose workflow framework (Altintas et al., 2004). The frameworks all differ in their setup and the design choices made. Table 3.1 summarises a few core qualitative characteristics of these frameworks, collected from personal experience by the authors and from the literature. The table is given as an illustration of the diversity of frameworks that exists, without pretending to contain a full quantitative comparison of the frameworks, which is considered to be beyond the scope of this paper.

Table 3.1: Characteristics of the examined modelling frameworks

FRAMEWORK	COMPLEXITY A	STANDARD FUNCTIONALITY B	IMPACT ON MODEL C	OPEN DEVELOPMENT D	TECHNICAL/ SEMANTIC E	FRAMEWORK TYPE F
Object Modelling System	Low	High	Low	Yes	Technical	Environment
TIME	Low	High	Low	No	Technical	Environment
OpenMI	Medium	Low	Medium	Yes	Technical	Specification
Common Modelling Protocol	Medium	Low	High	Partly	Technical	Specification
MODCOM	Medium	High	High	Partly	Technical	Environment
FRAMES	High	High	Very low	No	Technical/ Semantic	Environment
Kepler	High	High	High	Yes	Technical/ Semantic	Environment

A. The number of conventions the framework imposes upon the model for it to be compliant.
B. The amount of functionality included in the framework to support model integration, calculation and data analysis and presentation.
C. Framework invasiveness, the degree of dependency between the framework and the model code (Lloyd et al., 2011).
D. Denotes in how far framework development and use is controlled by an organisation, or is open for interested people to contribute to.
E. Type of model integration supported by the framework.
F. Whether the framework is only an interface specification, a specification with a default implementation, or a specification, default implementation including additional functionality, like a graphical user interface and model execution, analysis and visualisation components.

Studying the table of frameworks, it should be clear that even this subset of all the available frameworks varies tremendously in what they support and how they do so.

Developing a framework, starting with writing its specification from a set of requirements, providing a core implementation, and growing it into an environment with readily usable modules for e.g. database access, analysis, and visualisation is a time consuming activity. Choosing to invest and reuse a framework therefore partly becomes a strategic decision, and aspects like international recognition start to play a role. In this regard the four levels of model development and application ((i) specific single project use, (ii) used for a range of problems, (iii) well documented, published use in case studies, readily available for application, (iv) black box usage, accepted solution) defined by Argent, 2004 might as well apply to the framework development and maturity. When compared to other software framework solutions, e.g. Hibernate (http://www.hibernate.org/) for object-relational mapping between object classes and database tables, which can be and is widely being used in a black box fashion, model integration frameworks are clearly one or more steps away from reaching such level of maturity and acceptance.

3.2.2 OpenMI

This paper focusses on the use of OpenMI in a set of large multidisciplinary research projects in which models need to be linked. The OpenMI (http://www.openmi.org/) is a data and model integration framework, designed to take independent data and computing systems and provide a standard means of describing how time series are communicated between the systems. It has been developed from the need to answer integrated hydrological catchments management questions within the EU 5th framework programme project HarmonIT. The main objective of the HarmonIT project was to provide a widely accepted unified method to link models, both legacy code and new ones (Gijsbers et al., 2002).

As mentioned in Table 3.1 a modelling framework might exist as only a specification of a standard, independent of specific programming languages, providing a definition of its functionality and general operation, which can be implemented by others. For OpenMI this is referred to as the OpenMI Standard, and it is the core of the OpenMI work. A default (reference) implementation can also be provided together with the specification; this in general helps the adaptation of the framework. For OpenMI this is known as the OpenMI SDK (Software Development Kit). The core implementation can also be extended with additional functionality supporting, for example, model integration, model execution, data analysis, and presentation, creating a full model integration environment. An example of this is the OpenMI *OmiEd* simple model linking desktop application.

The OpenMI is constantly under development, under supervision of the OpenMI Association (OA), which aims to promote the development, use, management, and maintenance of the standard. To improve the adaption of the standard the OA at the moment provides reference implementations in the *.NET* and *Java* programming languages. The OA is an entirely open international group of organisations and people, with a small core team that supports, responds to, and is guided by a growing active worldwide user community.

It is a non-profit organisation and therefore depends on the willingness of its members. Organisationally, the OpenMI Association consists of a General Assembly, an Executive Committee (OAEC), a Technical Committee (OATC), and a Dissemination Committee (OADC).

The original version (1.4) of the OpenMI provided standardised interfaces to define, describe, and transfer data between software components that run sequentially, based on a pipes and filters architecture (Gregersen et al., 2007). The data definition concerns what the data are about (quantity) and where (element set) and when (time) it applies. Each component (a *LinkableComponent* object) has a meta-data description of its exchangeable data in terms of a quantity and an element set. Each unique exchangeable quantity is registered and published in an *ExchangeItem* object. Connections between *ExchangeItems* of *LinkableComponents* are defined by a *Link* and exist as a separate entity. Each Link can provide a number of *DataOperations*, functions performed on the data when it passes through the *Link* that transforms output data of one *LinkableComponent* into usable input data for the receiving *LinkableComponent*. *DataOperations* can perform e.g. unit conversions, aggregations or spatial mapping of point data to area data.

After the initial release of OpenMI version 1 in 2005 the OpenMI-LIFE project demonstrated its usability at the operational level on real world-scale problems (Schade et al., 2008), both hydrological and outside the hydrological domain (Van Ittersum et al., 2008), integrating agricultural and economical models from the farm field to the world scale). The OpenMI is seeing a significant increase in the number of applications leading to new requirements. For example, integration of non-model components such as the web-based hydrologic information system HIS (Goodall et al., 2011) or the decision support system AM-DSS was possible but required some effort. To improve the OpenMI and advance it to the next version, several European research institutes bundled activities and, after a modelling community review period, released the OpenMI standard version 2 in December 2010.

Some of the key ideas of the new version include:

i. A geographical representation for exchanged data approaching the common representation in the GIS-world. It is a step towards making OpenMI more compliant with the standards of the Open Geospatial Consortium (OGC).

ii. The Use of adapters, e.g. interpolating in space or time, to transform data into a requested form and of series connections of several adapters offering a piping and filtering mechanism. The new version considers such adapters to be special types of outputs, and a more straightforward replacement of the Link and *DataOperation* classes.

iii. Removal of the restriction to time-step based models enabling the integration of new kinds of models, e.g. *neuron networks*, in the future.

iv. Setting and varying boundary conditions for individual models for running comparative simulations, simplifying the use of the OpenMI in DSSs and tools for calibration, optimisation and data assimilation.

v. The introduction of a set of mandatory base interfaces and sets of optional interfaces has the aim of making the OpenMI fit for future requirements of integrated modelling. The current edition includes the extension for time and space-dependent components. Future extensions could support improved compliance with the standards of the OGC or request and deliver data in terms of dictionaries or ontologies.

Although the OpenMI standard does not dictate it, the current reference implementations are built around a single-execution thread (the smallest unit of processing that can be scheduled by an operating system) idea, requiring at least all the linkable components to be run from a single execution thread when calculating a chain of models. The linkable component itself is free to access a calculation core (usually the actual model) in any way it prefers, e.g. as a web service, which in turn allows the calculation core to use parallel or high-performance computing. In this regard, the focus of the OpenMI Standard and the current reference implementations is still to facilitate a high-level coupling of new and legacy models. The possibility of building a more parallel, cloud, or high-performance computing orientated implementation of the OpenMI Standard is still to be investigated.

The OpenMI is available under the terms of the LGPL open source licence (SDK implementations might apply the even less restrictive MIT open source licence) with the aim of easier dissemination. It is currently implemented in C# and Java and offers ways to wrap code from other languages. There are guidelines for migrating existing models to OpenMI compliancy in order to maintain approved legacy code. One of the more common ways to do so is to programme a wrapper *LinkableComponent* in Java or a.NET language that drives that legacy model and transforms input and output data into appropriate formats.

OpenMI was chosen as common strategy by Alterra, Wageningen UR in research projects targeting model linking, as (i) it is relatively lightweight, requiring few modifications to the models and works on the basis of well-defined and documented interfaces, (ii) it is based on an open standard, which is further developed by an multi-institutional organisation and which benefits from contributions from the larger research community, (iii) it provides flexibility to implement the standard in any programming language and to extend and adapt the implementation for the specific project, (iv) changes and additions to the standard can be proposed to the OpenMI-association for future developments and (v) it, as a multi-institutional organization, facilitates cooperation internationally and joint development of projects with OpenMI.

3.2.3 Projects

OpenMI has been applied as a modelling framework in several multi-disciplinary large projects that worked on integrated modelling, most of them funded by the European 6th Framework Programme (FP6). These projects operated in the disciplines of agriculture, land use, nitrogen cycling, forestry, hydrology, and economics.

Table 3.2: Characteristics of the examined projects

PROJECT	DOMAINS	PROJECT PURPOSE	HARDWARE / SOFTWARE PLATFORM	MODELLING LANGUAGES
SEAMLESS	Agronomy and economics	Model linking	Microsoft Windows based modern server, web browser based client (Flash Player needed).	Java, C#, GAMS, Adobe Flex
SENSOR	Land use	Results visualization	Microsoft Windows based modern server, web browser based client (Flash Player needed).	Java, Adobe Flex
EVOLTREE	Genomics and ecology	Model linking	Microsoft Windows based modern desktop PC.	Java, C++, Visual Basic
NITROEUROPE	Nutrient cycling and soil science	Results visualization	Microsoft Windows based modern desktop PC.	C#, Fortran
EFORWOOD	Forestry and forest ecology	Model definition and linking	Modern desktop PC, Java 5 compliant.	Java
Splash!	Hydrology	Results visualization	Windows desktop PC.	Java, Delphi

The SEAMLESS project (Ewert et al., 2009; Van Ittersum et al., 2008) developed a framework for an ex-ante integrated assessment of agro-environmental policies and agrotechnical innovations in the European Union. In this framework, a set of models operating at different spatial scales and from different disciplines are integrated into a model chain. The models are a cropping systems model, a bioeconomic farm model, an econometric estimation and up-scaling model, and a partial equilibrium market model. By linking field-farm-market models in a framework, changes in land use can be analysed at multiple levels through a selected number of economic, environmental, and social indicators, accounting for the impacts of farm responses that could not be analysed using only individual models as stand-alone tools. OpenMI v1.4 was used in SEAMLESS as a linking standard and SDK to link the different models together and to allow them to be executed on a personal computer in a joint run (Janssen et al., 2011). OpenMI v.1.4 was extended with the addition of semantic annotations for input/output exchange items.

The SENSOR project (Helming et al., 2008) developed a discussion support tool, the Sustainability Impact Assessment Tool (SIAT – Verweij et al., 2010), which facilitates multi-stakeholder discussions on the effect of different policy assumptions on multifunctional land use and its sustainability within different future images. SIAT allows the user to identify the geographical areas that are most sensitive to particular policies, identify regional differences, analyse causes, look at potential trade-offs, and perform all analysis dynamically (Potschin and Haines-Young, 2008). SIAT uses a small set of linked models to translate land use changes and sustainability indicators for environmental, social and

economic dimensions into land use functions (Paracchini et al., 2011; Pérez-Soba et al., 2008). The translation of policy to sustainability impact is done through a set of loosely coupled models which have been run for several scenarios to produce a large data store which is accessed by SIAT, and used dynamically by the linked Land Use Function model. The loosely coupled models are a macro-econometric model, a European forestry model, an agricultural model, and a land use allocation model (Jansson et al., 2008). In SIAT, OpenMI v1.2 is used as a linking standard and SDK to link different user interface modules.

EVOLTREE (http://www.evoltree.eu) is a network of excellence project that analyses the impacts of climate change on forest ecosystems. It sets out to link genomics and ecology to understand the evolution of diversity in terrestrial ecosystems. EVOLTREE develops methods to assess and forecast changes in biodiversity, structure, function, and dynamics of ecosystems and their services. To reach these objectives it uses experiments, genetic analysis, models and European wide datasets, and integrated modelling is only a small part of the EVOLTREE project. In EVOLTREE, OpenMI v1.2 was applied as a linking standard and SDK to link two models in different programming languages.

NITROEUROPE (Sutton et al., 2009) establishes an integrated perspective needed to quantify the net effect of N on the greenhouse-gas balance. It will advance the fundamental understanding of C-N interactions at different scales and deliver: process-based models, landscape-level assessments, European maps of C-N pools, Nr fluxes and NGE, and independent verification of GHG inventories, as required under the Kyoto Protocol. A part of the NITROEUROPE project is the development of INTEGRATOR, a modelling tool for European-wide scenario assessments of nitrogen budgets and greenhouse gas emissions. In NITROEUROPE, OpenMI v1.2 was applied to couple different inputs and outputs of the model to the user interface of the INTEGRATOR tool.

EFORWOOD (Lindner et al., 2010) built an assessment tool and procedure for environmental, social and economic sustainability in the Forest-Wood-Chain as a chain approach. Forest Wood Chains are defined as chains of production processes (e.g. harvesting-transport-industrial processing), which are linked with products (e.g. a timber frame house). The developed Tool for Sustainability Impact Assessment (ToSIA) calculates sustainability values as products of the relative indicator values (i.e. indicator value expressed per unit of material flow) multiplied with the material flow entering the Forest-Wood-Chain. Indicators are presented for the segments or for the complete chain. Sustainability is determined by analysing environmental, economic, and social sustainability indicators for all production processes along the chain. In ToSIA, OpenMI v1.2 links many different small models together, each model representing a single process or step in the calculation.

Splash! (Wachowicz et al., 2003) is a Sim City-like game for teaching aspects of water-management. The player takes the role of a super water-manager that has to arrange all the spatial activities (e.g. farming, industry, cities) in the game area properly to assure clean

water and keep all virtual stakeholders (citizens, employers, farmers, environmentalists, etc.) happy. Building dikes, water towers, and strategic zoning are some of the tools the player can use, naturally within all kinds of constraints, like a limited budget and the occasional simulated flooding. The game area is modelled according to an existing (Dutch) area and simplified models are used to simulate the results of the actions a player takes. Splash! is primarily intended as a teaching instrument. OpenMI v.1.2 connects the game engine to the user interface in Splash!.

3.2.4 Method

To evaluate the use of OpenMI in the mentioned projects and assess its value for model integration undertakings, two half-day workshops were organised, one with software developers and the other with domain modellers. Software developers have a training in Computer Science, and are practised in building complex applications for personal computers by adopting the latest innovations from the computer science domain. Domain modellers have been trained in specific domain (e.g. agronomy, economics, forestry) and focus on conceptual model development, implementation, and testing as done in a specific scientific domain, with often one known technology that works. In the software development workshop, we invited six software developers who were involved in the projects and worked with the modellers on the integration of their models through the use of OpenMI. For each project, we asked the developers to give a short presentation on the goal of the project, the role OpenMI played in the project, the strength and weaknesses of OpenMI based on the experiences in the project, the lessons learnt and the suggestions for improvements to OpenMI.

After the presentations a discussion was held about the outcomes, particularly focussing on the noted strengths and weaknesses, lessons learnt, and suggested improvements. The information collected from this and the presentations was gathered and combined into a first concept of this paper, which was then handed to the modellers involved in the projects and whom we invited for a second workshop.

In this second workshop, five modellers were asked to respond to both the concept of the paper and to our initial findings and conclusions, approaching it from their own perspective. For the most part, it was a round-table discussion that provided lots of feedback and triggered interaction between the modellers. Typically, the software developers present at the first workshop all knew each other and worked together on one or more occasions, while for the modellers it was their first time to interact about the subject.

3.3 Results

3.3.1 Roles of OpenMI

Based on the feedback from the developers and the modellers gathered during the workshops, two distinctive ways were used to apply the OpenMI in the projects: (a) using models directly and modifying the model source code to make it OpenMI compliant (Figure 3.1), and (b) building a wrapper component to provide the OpenMI compliance while leaving the model largely untouched (Figure 3.2). Using the model directly from a *LinkableComponent* allows for more detailed integration, but usually takes more time to implement and requires the use (in the model) of the same or at least compatible programming languages. For example, if you are using the Microsoft .NET environment, OpenMI SDK and OpenMI graphical editor (OmiED), it is easy to directly integrate a model written in C# or Delphi for .NET, but harder to integrate a model written in Scala or Fortran. In such a case, a wrapper of some kind is needed to cross the programming language barrier. The wrapper approach is also useful to make models comply with the OpenMI standard in regards to data definitions, exchange formats (e.g. from in-memory data exchange to file or database-orientated exchange), and model execution sequences. Building generic wrappers makes it possible to reuse this often required functionality.

Option A was used by three of the projects, option B by two, and the SEAMLESS project applied both methods to integrate models. A variety of models from different domains were linked with the use of OpenMI in the projects, with a large diversity in the spatial resolution the models operated on (e.g. from farm field to world market scale), the size of the models (from EU market equilibrium model to very small wood processing steps with many feedback loops), and the programming language used for the model (Java, C#, GAMS, Visual Basic, NSM, CAPSIS – the latter two being complete modelling environments that were made OpenMI compliant). Furthermore, the time dimension in these projects played a less significant role than in typical OpenMI usage, where a whole composition of linked models performs calculations in a time-step-based fashion. The Java version of the OpenMI standard (1.2 and 1.4) was used and matching Java OpenMI SDKs (Software Development Kits) to help with the implementation of models and systems. Where needed, innovative solutions were added to the SDKs and later presented to the OpenMI Association for consideration.

The Nested Systems Modelling (NSM - Van der Werf, 2009) and the Computer-Aided Projection of Strategies in Silviculture (CAPSIS – http://capsis.cirad.fr/) that were used in the projects and linked with other models in itself were modelling environments that as a whole were made OpenMI-compliant. A very large initial development investment is required, with the reward of later being able to use all kinds of OpenMI-compliant models in the existing modelling environment. In case the programming languages match (e.g. CAPSIS, written in Java could use the Java SDK to make it compliant with the Java

OpenMI standard interfaces) it takes less development time than when they do not (e.g. NSM written in C++) and more complex technology is needed to bridge the differences in programming languages, virtual machines, etc. Using Microsoft COM, .NET interop or Oracle's Java Native Interface (JNI) has some possibilities, but these solutions are hard to debug to find the cause of errors due to the multitude of programming environments. In this case, using a network interface as a clean boundary between (too) different systems might be a more manageable solution. This idea was used in the Splash! project to integrate the completely different game engine (written in the Delphi programming language) with a Java-based simple OpenMI model linking and execution system (Figure 3.3). Plain network sockets were used for the interface and communication between the two parts, keeping each separately implementable and testable.

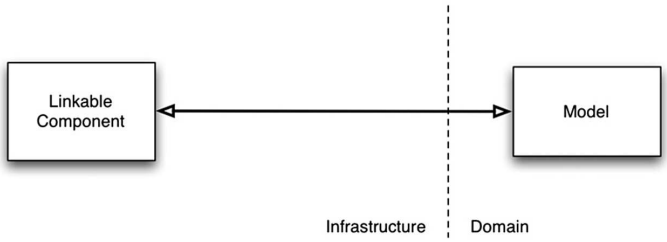


Figure 3.1: Option A - Applied by SENSOR, EFORWOOD, SEAMLESS and Splash!

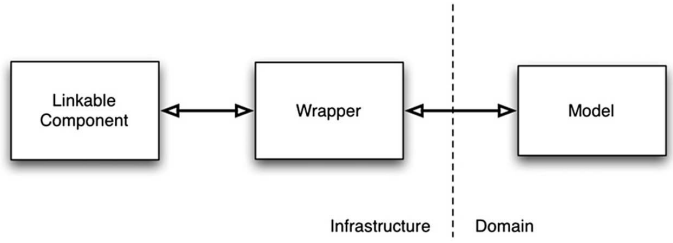


Figure 3.2: Option B - Applied by NITROEUROPE, EVOLTREE and SEAMLESS

3.3.2 Detailed case: OpenMI in SEAMLESS

As explained in Section 3.2.3, OpenMI v1.4 was used and extended in SEAMLESS to link a cropping system model, a bioeconomic farm model, an econometric estimation model and a partial equilibrium market model. In the SEAMLESS modelling framework, each model is linked to SeamFrame through wrapper components, which are implementations of OpenMI components (Figure 3.4). Each wrapper prepares the data for the model in input files, runs the model by calling on an executable, and reads the results from the output

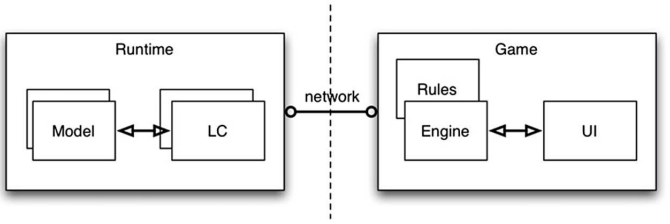


Figure 3.3: Option C - Applied by Splash!

files. Models were implemented in C#.Net for the crop model, in Java and GAMS for the farm model, in GAMS for the upscaling and market model, while SeamFrame was built in Java. In between models data is managed in SeamFrame by executing a list of models in a chain (i.e. SeamChain in Figure 3.4). The shared definitions of data types are captured in the SEAMLESS ontologies (Athanasiadis et al., 2009; Janssen et al., 2011) and are used in SeamFrame and model wrappers (Figure 3.4), but not in the models themselves. In this way, models are separated from each other, through wrappers and SeamFrame, and from the knowledge layer describing the exchanged data types as captured in the ontologies.

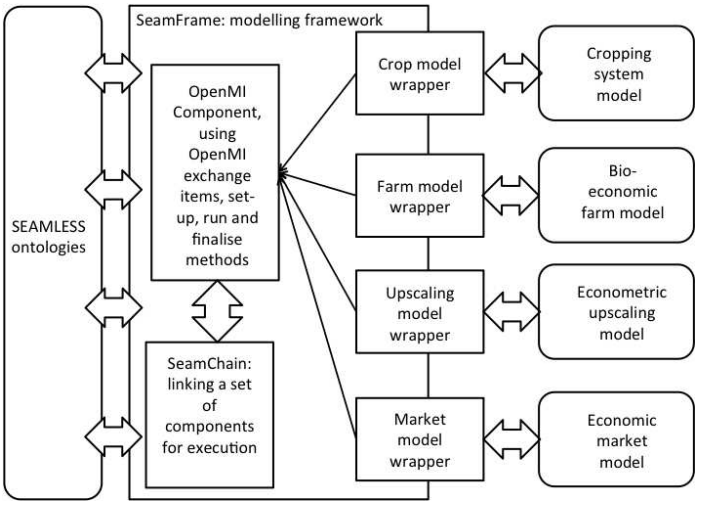


Figure 3.4: Use of OpenMI in the SEAMLESS Modelling Framework

In the execution of the model chain, if all four models are included, the use of the modelling framework including OpenMI takes little execution time compared to the models themselves (Table 3.3). This is related to the nature of the models and the application.

The crop model needs to be executed many more times for one sensible application than the market model, leading to a long total execution time, even if the execution of one crop model requires little time. Only after the market model, considerable time is required to persistently store a multitude of results generated by this model. Some of the models included in this case are optimisation models that sometimes need many iterations to arrive at a solution, requiring a long computational time. This time division between models and the modelling framework could be different if models are considered from another domain. This case suggests that more efficiency gains in time can be made by considering the models, and not as much the framework.

Table 3.3: Estimation of time required for a full chain run of the models in a typical application for SEAMLESS.

Model Type	Number of runs required in typical application	Time (sec) required per run	Time (sec) in preparing inputs	Time (sec) in storing outputs
Crop model	900–1000	30	0.5	1
Farm model	50–60	60–1200	2	1
Up-scaling model	1	300	5	10
Market model	1	3600	20	400

3.3.3 Strength and weaknesses

Table 3.4 gives an overview of the aspects mentioned in the workshops that are considered a strength or benefit of using the OpenMI, including both the OpenMI standard (the interfaces, concepts and approaches to model linking) and an OpenMI Java or.NET Software Development Kit (SDK). As described in Section 3.2.3, the projects involved and the people working on them all used some kind of soft-linking – manual and ad-hoc linking of models through using output files of one model as inputs to another – so mentioned strengths and, in the next table, weaknesses are based on comparing this to the OpenMI.

All comments mentioning the same aspect were grouped and the number of projects for which the comment applied is listed in the second table column. An aspect that is mentioned more often (by more projects) is considered to be a more relevant point, and hence a better argument for considering or discarding the OpenMI as a suitable solution for a project. It can be observed that for the mentioned advantages, the first 1 to 5 are more general in nature, and only 6 and 7 directly relate to the OpenMI.

In addition to strengths, naturally, several weaknesses were also mentioned. These are summarised in Table 3.5, using the same grouping strategy and layout as in the previous table.

Table 3.4: Strengths of the OpenMI framework

	STRENGTH	#PROJECTS
1	Open source standardisation; provides clear and fixed definition on how things are supposed to work; potential for interoperability.	5
2	Separate interfaces from implementation (increases software maintainability); ability to use (wrap) existing models.	3
3	Re-use existing OpenMI framework experience and knowledge.	2
4	Level of available documentation.	2
5	Solution for infrastructure (not a project goal to develop it).	2
6	Ability to operate over a network; enough flexibility in the framework specification; works well for small and large models.	2
7	Quick support feedback to questions.	1

Table 3.5: Weaknesses of the OpenMI framework

	WEAKNESS	#PROJECTS
1	Regarded as complex or overkill by modellers.	5
2	Missing (expected / promised) environment and standard shared modules, e.g. for data analysis and visualisation, multi-thread execution, programming language bridges.	5
3	Missing required features: OGC support, qualitative data exchange, raster data, multi-part geometries, loops, N:M connections between inputs and outputs, support for semantic integration.	4
4	Cumbersome to use, requires work-around for models outside the hydrology domain.	3

For the weaknesses most are obviously directly related to the OpenMI SDK, which, as a conclusion, at the time of the projects was too immature to cover all the wishes of the modellers out-of-the-box. Only argument 4 relates to the OpenMI standard, which at the applied version 1.4 was still very specific for the hydrology domain and needed adaptation and flexibility in use to make it fit to a broader environmental scope.

Although many strengths and weaknesses were mentioned during the two workshops, some clearly distinct groups of comments became apparent. The participants (both developers and modellers) liked the use of a standard and the benefits it apparently (perhaps supposedly) offers, which are clarity in definitions, available documentation, support and samples to work from, interoperability, improved software maintenance, reuse of experience and skills. Having such an infrastructure in place is a benefit and typically not a primary goal of an integrated environmental modelling project to build it but instead to use it. The weaknesses of the current OpenMI, in this regard, are that it is regarded as complex, cumbersome to use, and lacking features or standard functionality (that could provide the added value to researchers working on model integration). An often-encountered misunderstanding is the difference between OpenMI as a standard and OpenMI as an implementation of the standard in a programming environment.

As a final result of the workshops Table 3.6 gives an overview of the improvements for OpenMI, proposed by the participants. Most notably are (i) the clear wish for something to help with an easy and fast way to start with using the OpenMI, and (ii) to better communicate the short-term benefits of OpenMI instead of the long-term strategic goals.

Although all these improvements related to the use of OpenMI have been mentioned, most are very generic in nature. When introducing a model integration platform to a project, it should not be a burden but as easy and as quickly to use as possible. Besides that, it should not be overhyped to get it accepted into a project or forced upon a project where it does not serve a purpose.

Table 3.6: Suggested improvements for OpenMI

	SUGGESTED IMPROVEMENTS	#PROJECTS
1	Quick start manual.	3
2	Do not hype / over-market it; better communication.	3
3	Do not force it upon modellers when it does not serve the project.	2
4	Graphical editor.	1
5	Add semantic layer to standard.	1
6	Increase ease of developing and testing model components.	1

3.4 Discussion

This section presents an evaluation by the authors based on the results of the workshops, summarised in the tables in the previous section, combined with their general knowledge and experience from working in the fields of environmental integrative modelling, agricultural modelling, and software engineering. The arguments and conclusions written here should help fellow scientists; both integrative modellers and software developers, make more informed choices and avoid similar potential pitfalls. In addition to that, for software engineers working on model integration platforms, there are clear suggestions for areas to focus on, which will increase the likelihood of adapting a framework.

The interviewed modellers and developers subscribe to the idea of using standards for integrative modelling and the potential benefits this presents. Only the definition of the standard is not enough; a full implementation is expected that handles all the boilerplate work and provides all the right tools and components. Domain-orientated projects are not setup and properly budgeted to contribute to framework development besides the core content of the project itself, which was the case in the projects reviewed here. With the standard specification available, benefits are already experienced, but the standard functionality needs to become available as well.

A further cause of tension is the difference between building a model with a stand-alone purpose and building a model suitable for linking to other models. The latter is usually not the primary goal of the modeller working on the model and neither of the software developer working on the framework. Making a model linkable clearly is an effort that falls between two stools. Similar tension exists between developing models and integration solutions specifically for a project versus developing them in a more generic way. Making a strong (supra-)institutional strategy is helpful, which must also be supported by strong project acquisition goals.

A team effort between software developers and modellers to create an integrated model with use of a software framework requires the software developers to learn about the modelling and the domain context, and on the other hand the modellers need to pick up some of the software engineering science and principles that they might not be accustomed to. Common best practices of the software engineering community, such as using version control, issue management, unit tests, continuous integration, as well as the Unified Modelling Language (UML) and domain modelling (software), can be relatively new to the modeller (Knapen et al., 2007; Verweij et al., 2010) and account for the burden of having to use the framework. Thus, the complexity of model linking in a rigorous and repeatable solution might overwhelm the modeller and software engineer when confronted with the multitude of tools and approaches from software engineering and the complex data types, large amounts of data, and scaling issues from the modelling domains.

Advantages of using an existing and standards-based model integration framework are (i) the short-circuiting of a lot of discussions, commonly when working with researchers, since the standards have to be followed. The alternative is to spend extra resources on improving and adapting existing functionality; (ii) it provides the opportunity to quickly start with integration, run the model-chain and start acting on the results by improving the models and making them more suited for the integration; and (iii) the adoption of a standard proves to lead to better documentation and construction of the model making them more comparable and easier to follow for researchers from other domains. The models do not automatically become interchangeable, in practice a project tends to build a wrapper layer around each model and a layer on top of a framework for the purpose of integration and matching the framework set up to the specifics of the project (i.e. modelling languages, model scales, desired end-use). Such layers can at a later stage be used to improve the models and the framework itself.

Considering the large diversity of projects, domains (i.e. agriculture, land use, forestry) and types of models described in this paper, the OpenMI standard can meaningfully be applied outside the hydrology sector from which it originates. Some of the extensions and adaptations required to fit project needs (i.e. more flexible data types, better support for non-time-stepping models) and mentioned framework weaknesses (i.e. too complex and cumbersome to use outside the hydrology domain) are already being addressed in version 2.0 of the OpenMI standard specification (Donchyts et al., 2010). In addition, in its SDK implementations, which focus more on making its usage more lightweight and less invasive for the models.

From the use and evaluation of OpenMI across the projects, a number of recommendations can be made to modellers or developers interested in using OpenMI as their modelling framework. First, a sufficient understanding of software development and its principles must be available or developed before making a serious effort in model integration. Second, if models with an existing (large) code base are integrated, OpenMI is a relevant solution

as it intends to work with wrappers around the models, not requiring changes to the model. Third, the commitment of both the modeller and the software developer is crucial to identify and implement the often complex data types exchanged between models, probably using any modelling framework, not just OpenMI. Fourth, different implementations of OpenMI exist, allowing for some flexibility to choose the implementation best fitting the model characteristics (i.e. language, operating system). Fifth, use of OpenMI is most useful if integrated models can be used in different projects, ensuring the maintenance and further development of integrated models with new subparts or extensions in other subdomains.

Overall, OpenMI is a useful operational solution for integrating existing and new models across domains and scales in a technical sense, often based on wrapper development and in close interaction between modellers and software engineers. Future attention has to be on developing and sharing standard functionality for use with model chains, achieving easy re-use of available OpenMI compliant models, and moving towards simple, unobtrusive, and ubiquitous uses of this standard in the field of environmental modelling.

Acknowledgements

The authors appreciate the help and support of many scientists in achieving model integration in research projects. The authors also thank Koen Kramer, Gert-jan Reinds, Bert van der Werf, and Martin van Ittersum for their valuable comments and ideas. The work presented in this publication is supported by the Knowledge Base Research Grants of the Dutch Ministry of Economic Affairs, Agriculture, and Innovation. The development of OpenMI version 2.0 has been funded by the members of the OpenMI Association, as well as the European Commission through various 6th Framework programmes and the LIFE Environment programme. Many of the projects mentioned in this document have also been funded by the EU 6th Framework programmes.

❄

Chapter 4

Connecting Systems

This chapter is based on:

M. Knapen, A. De Wit, E. Buyukkaya, P. Petrou, D. Paudel, S. Janssen, and I. Athanasiadis (2025). "Efficient and scalable crop growth simulations using standard big data and distributed computing technologies". *Computers and Electronics in Agriculture* 236, 110392. DOI: https://doi.org/10.1016/j.compag.2025.110392

Abstract

The digitisation in agriculture has led to an explosion of highly detailed data generated, offering opportunities to further optimise resource use in food production systems. However, managing and processing these growing data volumes presents significant challenges. This study investigates the suitability of standard big data and distributed computing technologies with a crop yield forecasting case study, and benchmarks performance and scalability of storage and compute. To that end a prototype system leveraging the Apache Spark big data analytics framework and using the WISS-WOFOST crop growth simulation model is assembled and evaluated for its efficiency and scalability when running large numbers of simulations using distributed computing on commonly available infrastructure. Existing data for maize and winter wheat, as typical summer and winter crops, is prepared for distributed storage and processing and used to measure the performance of the system on clusters of increasing sizes, from small Kubernetes Cloud deployments to large HPC configurations. Specific attention is paid to the aggregation of the grid-based simulation results to larger administrative regions for follow-up analysis and reporting. Our results demonstrate that the selected standard big data and distributed computing technology simplifies the application of distributed processing and storage, making the related trade-off between runtime and costs more attainable. By increasing the distribution of our system 64 times and the total number of cores used 45 times compared to the baseline, we obtained a 99% reduction in simulation processing time and a 95% decrease in the aggregation time of the simulation results, making detailed forecasting for large areas more tractable. However, distributed implementations remain inherently more complex than conventional ones. As such, the construction and use of distributed systems will continue to be a challenge for agricultural agronomists and agricultural data scientists.

4.1 Introduction

The ongoing digitisation of agriculture provides increasing amounts of data that can be used to further improve our food production systems, including optimising resource use on the farm with precision agriculture, forecasting regional and global yields, helping us to adapt to the effects of climate change and allowing consumers to make better informed decisions about their food purchases by tracking and providing sufficient information (Parra-López et al., 2024; Wolfert et al., 2017). All of these require attention to how increasing amounts of data are stored, managed, and processed, both in operational systems and for research purposes.

Higher demands for data processing and storage capacity can be handled in the first place by upgrading computing equipment with more storage space, more memory, and faster processors. This is referred to by the term "scaling up". To make the most efficient use of the available hardware, concurrent processing can be implemented, utilising all CPU cores in the system to their maximum. However, eventually all this will reach system limits, and the use of a form of distributed computing (known as "scaling out") has to be considered in order to still be able to perform all required processing in a timely fashion (see Hennessy and Patterson, 2011).

Distributed computing refers to the use of a cluster of computers, typically with highly available resources (processing cores, memory, and disk space). The main computational task can then be divided and solved using all computers available in such a cluster, with a final task that collects all outputs and integrates them to produce the end result. Due to the inherent higher complexity of distributed systems, the initial (good) tendency usually is to avoid using such solutions, stick to familiar single computers, and attempt to fit the computational job. Or, not being familiar with the existing available IT technologies, bespoke systems using multiple dedicated computers are constructed (such as, for example, in Kim et al., 2020) which are highly dependent on specialised software and tailored servers, usually not very fault tolerant and potentially a maintenance nightmare.

Fortunately, custom individually managed servers that require personal attention can now be replaced by ephemeral commodity servers either located on-premises, for example, as part of local Kubernetes cluster (http://kubernetes.io) or HPC (high-performance computing) facilities, or hosted remotely ('in the Cloud'), e.g. Microsoft Azure, Amazon Web Services, or the Google Cloud Platform. Making use of such computing facilities still requires breaking down the total computational workload into a number of smaller tasks that can then be processed in parallel. This scheduling, or orchestration, software can be custom developed, for example, by accessing the Kubernetes Control Plane that allows the dynamic creation and deletion of nodes in the cluster (if administrators allow) as done by Kim et al., 2021, or by programming master-worker node implementations using internal networking and a central database for distribution of tasks, as described in Li et al., 2023b.

Alternatively, standard big data and distributed computing technologies can be leveraged, such as the currently well-known open source frameworks Dask (http://dask.org), Ray (http://ray.io), and Apache Spark (http://spark.apache.org).

In this paper, we investigate whether such a standard distributed computing solution can be successfully applied to a classical use case from the agricultural domain, namely that of crop yield forecasting. This is a key component of crop yield monitoring systems, which are important tools for agricultural monitoring (e.g. for anomaly detection and early warning) (Fritz et al., 2019). They are critical to informing stakeholders on the current outlook on crop production and provide support for policies on market intervention and import/export regulations. A few examples are the international Agricultural Market Information System (AMIS, http://amis-outlook.org) of the Food and Agriculture Organisation (FAO), and the GeoGLAM Crop Monitor (https://cropmonitoring.org). A variety of crop yield monitoring systems exist, some based on field visits and in situ observations (e.g. USDA system (U.S. Department of Agriculture, 2012)), some relying on remote sensing imagery (e.g. FEWS-NET (Ross et al., 2009), NASA Harvest (Whitcraft et al., 2020), CHARMS (Huang et al., 2018)) and a final category applying deterministic crop growth models, often combined with other data sources including remote sensing data (e.g. EC MARS (Lecerf et al., 2019)). This last category of crop yield monitoring solutions ingests data on crop, soil and agro-management, and historical weather data, as well as ensemble weather forecasts. Next, a cropping systems model is applied to provide estimates of various crop variables, such as phenology and biomass, and to provide a forecast of the expected crop yield at the end of the season. Monitoring systems are traditionally implemented on a consistent geographical grid of 50×50 km or 25×25 km, on which all data are prepared and different information layers are intersected. With the advance of digitisation in agriculture, more data is becoming available with increasingly detailed spatial resolution. Examples of such data products are the AgERA5 weather data set (Boogaard and der Grijn, 2020) at a resolution of 0.1 degrees, and the SoilGrids soil database at a spatial resolution of 1 km (Poggio et al., 2021). Crop monitoring systems would be more useful if they were upgraded to provide outputs with higher spatial resolution utilising those new data sets (Paudel et al., 2023). This increases their usefulness as the outputs become relevant for different user groups which often require data on cropping conditions for smaller spatial entities (e.g., watersheds and counties). For example, the outputs of such systems could be used for index-based insurance (Afshar et al., 2021) and real-time advisory services for farmers and extension services (Hack-ten Broeke et al., 2019). However, such an upgrade to high resolution significantly increases the computational requirements for operational systems as the number of unique simulation units grows. How to address this, preferably with existing tools, is an important research question.

Thus, in this study, we investigated, as the main research objective, *whether a commonly used distributed computing framework could successfully be applied to run simulations at scale using a numerical crop simulation model, while benchmarking both performance and*

scalability of the system on data sets for two types of crops (maize and winter-wheat, as typical summer and winter crops), and deploying it on computer clusters of various sizes, using both Cloud and HPC configurations. However, to reach this objective, we first needed to look at an existing system and adapt its key data management and processing steps to make use of distributed technologies. Based on our familiarity with the WOrld FOod STudies (WOFOST) cropping system model and its operational use in the MARS Crop Yield Forecasting System (MCYFS) (De Wit et al., 2019), these were selected as the basis for this work.

In section 4.2 we describe an implementation of a distributed system that embeds a WOFOST cropping system model as a data transformation in Apache Spark, making use of aspects of declarative (functional) programming. This includes key aspects for data management in distributed computing, such as the data input and output schemas, data denormalisation, data serialisation, and data deserialization, what distributed computing entails, and how we used it to run the existing crop model, as well as for aggregating the outputs of all model runs. Having this prototype system in place, we used it to run the crop simulations for the maize and winter-wheat data sets, on various distributed computing configurations. section 4.3 documents these benchmark experiments and their results, which we further discuss in section 4.4, with the final conclusions in section 4.5.

4.2 Methodology

4.2.1 Overview

To meet the research objective presented in the Introduction, we chose to use the gridded implementation of the WOFOST cropping system model (De Wit et al., 2020a) within the European MARS crop monitoring system (Van der Velde et al., 2019) as a reference for prototyping and benchmarking a distributed technologies-based system as a case study.

Figure 4.1 shows an overview of the processing steps in the prototype system and already mentions some of the distributed technologies used in the prototype (such as the Apache Spark framework), which will be described in detail later. As part of the implementation, we needed to find effective solutions for (i) managing the large input data for crop growth simulations using distributed storage technologies (box 1 in Figure 4.1); (ii) efficiently running a numerical crop growth model on a computer cluster (box 2 in the same figure); (iii) collecting the results of large numbers of both successful and failed simulations from all computers in the cluster (also in box 2); (iv) applying post-processing operations on the vast amounts of simulation outputs to obtain the final results (box 3). More details about these technologies and the solutions chosen are described in the following subsections.

Our prototype leaves out or simplifies steps at the boundary of the system, i.e. it uses data from the existing MARS system, and has no user interface components for control and visualising outputs. To construct an operational system, these will have to be added; however, they are not needed to experiment with the processing of the core crop simulation model and to perform benchmark measurements.

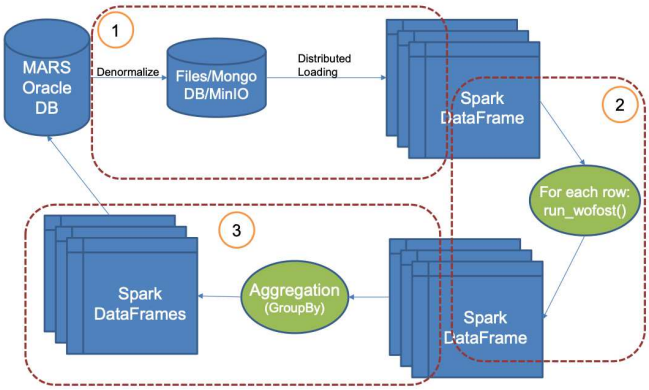


Figure 4.1: Overview of the crop simulation processing steps. (1) Extraction, transformation (denormalisation), and loading of data into the Spark analytical framework, which represents it as a `DataFrame`. (2) Distributed and parallel running of crop simulations for each row of the `DataFrame`, producing results as a new `DataFrame`. (3) Using Spark SQL commands to calculate aggregated outputs.

4.2.2 Case study

A clear use case for the sketched prototype that can scale out crop simulations is the computation and use of the resulting crop simulation outputs for regional crop yield forecasting. Examples of such systems are the mentioned European MARS Crop Yield Forecasting System (MCYFS) (Van der Velde et al., 2019) and the CRAFT system (Vakhtang et al., 2019) which were both designed to provide estimates of crop production at regional level during the cropping season. Similar systems are in place in other parts of the world, such as CGMS-Morocco (De Wit et al., 2013; Lahlou, 2018).

Complex systems like MCYFS or CRAFT typically segment the spatial domain into small spatial units for which all input variables are assumed to be homogeneous. Next, a crop simulation model is applied to each spatial unit, and its output is collected. The reason for this approach is the non-linear response of a crop model to its inputs which implies that simulations must be done at the lowest spatial level, followed by aggregating the model outputs towards higher levels such as grids or regions.

Currently, the MCYFS operates at a spatial resolution determined by the intersection of the weather grid (25×25 km) and the soil map, which provides homogeneous grid/soil units. For a crop such as winter wheat, the system employs 150,000 individual soil/grid combinations that have to be simulated individually. Given that the system simulates about 20 types of crops operationally, this will add up to approximately 3 million individual simulations with the included model (WOFOST, further described in Section 4.2.3). These simulations must be repeated every 10 days to take into account the latest weather conditions.

Besides the computational burden that comes with running WOFOST on small individual units, handling results from all those individual units also requires quite some attention. The WOFOST simulation results at the lowest level are of little use for yield forecasting and visualisation. Therefore, results are aggregated to grid and regional levels, each following a different aggregation scheme. Aggregation from the lowest level towards grid level is performed by computing an average of the simulated variables weighted on the relative area of each soil type within the grid. Aggregation of grid level results towards the lowest level regions is performed using the area of arable land for each grid as a proxy for crop area, while aggregations from the lowest level towards higher levels are carried out by crop area estimates obtained from EUROSTAT (http://ec.europa.eu/eurostat). Moreover, at each level of aggregation, a climatology is required that is used to produce maps and charts showing the current conditions relative to the long-term statistics.

In the current implementation of MCYFS, all crop simulations are done on a single compute server that has a multi-core processor. The source code and data retrieval of the model have been optimised for performance on this limited infrastructure. The current distribution of simulations across the processor cores is done by splitting the spatial domain into tiles, which works in practice but gives little flexibility: tiles which consist of fewer simulation units (e.g., which contain few crop areas) will finish quickly but cannot pick up tasks from other tiles (i.e., there is no "work-stealing" between tasks implemented). In addition, results from individual crop simulations are first written to files and then loaded into a relational database using dedicated data loading tools, which takes a considerable amount of time. Finally, the aggregation of simulation results is carried out using database procedures that compute the weighted averages for grids and regions, which is demanding, although something that databases excel at. Finding solutions for the challenges mentioned above will be critical for scaling crop monitoring systems towards higher spatial resolutions.

4.2.3 Embedding a WOFOST model

The WOFOST cropping system model has been used in the MARS crop yield forecasting system since the early 1990s (De Wit et al., 2019). It is a core component used to determine the impact of weather conditions on crop growth for the major arable crops in Europe and other areas of the world. The model computes crop growth and development in a

biophysical and process-based way and summarises the status of the crop through a set of core state variables: phenological development, biomass in various plant organs, leaf area of the plant canopy, and its interaction with the soil through soil moisture. The changes of these variables from day to day are computed based on the underlying processes such as photosynthesis, respiration, leaf dynamics, evapotranspiration and how they are influenced by weather.

Crop growth simulation models themselves, such as those used in this study, are difficult to further parallelise efficiently because of their internal use of interrelated state variables calculated in time steps. Moreover, many of such existing models were not developed with parallelisation and distributed computing in mind and contain legacy programming and design constructs that make a switch difficult. For example, Jang et al. (2019) developed a parallel computing framework to run a spatialised version of the EPIC model. However, their approach still relies on file-based input/output which severely limits distribution over multiple computing nodes. In addition, the structure of the model is such that simulations are limited to using multiple CPU cores on a single computer, rather than using all CPU cores in a cluster of multiple computers. Similarly, Alderman (2021) developed a gridded version of the DSSAT-CSM model that has comparable limitations on scalability.

Running a single WOFOST simulation for a given location, crop, and year can be carried out in less than 100 milliseconds using an efficient implementation on modern computer hardware. However, the MARS system is a spatial implementation of WOFOST which is computationally demanding overall. Given the strong non-linearity in crop models, a spatial implementation of WOFOST must adhere to the principle of simulation at the lowest level first, followed by performing output aggregations in time and space. Therefore, each unique combination of the weather grid, crop mask, soil map, and agro-management must be simulated separately, which generates a large number of unique simulation units.

Developments within the MARS system have steadily increased computational requirements. These included a decrease in the size of the meteorological grid from 50×50 km to 25×25 km, extension of the area to be monitored, and inclusion of additional crop types. Moreover, the inclusion of ensemble weather forecasts strongly increases the computational load because model ensembles need to be calculated instead of deterministic model runs. Efficiently handling such a computational load requires an optimised implementation of WOFOST as well as a data infrastructure that can handle a large amount of input data and output data in a performant way.

Recently, the 7.2 version of WOFOST (De Wit et al., 2020a), originally written in FORTRAN, has been reimplemented in the Java programming language, focussing on efficiency, performance, modularity and portability. This edition is also known as WISS-WOFOST and consists of a lightweight framework (WISS, for Wageningen Integrated Systems Simulator) (Van Kraalingen et al., 2020), and a component-based implementation of the crop simulation model itself. An important design principle for this edition was that

the model should be free of side effects as much as possible and can be run completely in memory. Furthermore, the source code has been fully aligned with the WOFOST implementation, which is part of the Python Crop Simulation Environment (PCSE) (De Wit, 2021).

The only remaining side effects of WISS-WOFOST are exceptions (errors when running the simulation) and log messages. These still make it an *impure function* (see Section 4.2.4) since it is not completely referential transparent, but both types of side effects can be handled with relative ease. This functional programming-style implementation of WOFOST together with the versatility and efficiency of the Java programming language greatly increases the opportunities for using WISS-WOFOST in computational challenges and big data applications. Java programmes are ultimately compiled into Java bytecode and executed on a Java Virtual Machine (JVM), which then continuously performs dynamic analysis and optimisations of the running code. Some other programming languages are supported by the JVM as well, and language interoperability is then a given. This is particularly convenient for embedding WISS-WOFOST into a framework such as Apache Spark, which is written in Scala, one of the other JVM languages. However, the interoperability of Spark is by no means limited to all JVM languages. It also supports the frequently used programming languages in data science, Python, and R, and it includes functionality to access command-line based models and applications (e.g. compiled C++ or Fortran code) via Linux pipes and interprocess communication.

The input data required by the spatial WOFOST implementation used in the MARS crop yield forecasting system is stored in a relational database management system (RDBMS). The use of a RDBMS is necessary because of the normalised structure and the many relationships that exist between the different input data types (crop, soil, weather) and the WOFOST output data for each unique combination. However, over the course of three decades, the database schema behind the spatial WOFOST model has grown in complexity to accommodate new features and options. This complexity has a negative impact on the database performance for supplying the input data for WOFOST. Typically, many tables have to be traversed and joined in order to obtain all the required inputs for a WOFOST crop simulation, particularly those related to model parameters and crop calendars. Moreover, a RDMBS and the server it runs on can only handle a limited number of simultaneous connections, making them less scalable.

Fortunately, despite its complexity, most of the input data in the spatial WOFOST RDMBS are static: they do not change during the course of a cropping season. Only weather observations are appended based on new daily observations. This provides an excellent opportunity to convert the WOFOST input data stored in the RDBMS into a temporary format that is more suitable for distributed processing. In the following, we will describe the approach we have implemented for this.

Finally, the WOFOST model produces a time series of output variables for different model scenarios (e.g., potential or water-limited production) for all unique combinations of grid, soil, and crop. However, in practice, analysts never use the WOFOST results at the lowest level of simulations. Instead, aggregated values are preferred at the grid or regional level because they are easier to handle and visualise. Currently, all WOFOST simulation results at the lowest level are loaded into the RDBMS and the spatial aggregations are carried out using SQL procedures. This step is time-consuming because all data have to be loaded first before aggregation can start. We have therefore experimented with an alternative approach in which aggregation of results is already carried out on the Spark `Dataset` that stores the WOFOST simulation output. This results in a much smaller amount of data that have to be loaded into the RDBMS.

4.2.4 Functional programming

The design of computing systems is closely related to the programming languages that control them, as each language is built on a computational model that shapes its paradigm and programming style (Backus, 1978). Two major paradigms are imperative programming and declarative programming (Roy and Haridi, 2004).

Imperative programming languages, such as C, Java, and Python, are based on the Von Neumann computer architecture (described in 1945) and operate through sequences of commands that modify the state of the programme. Although modern imperative languages have begun incorporating functional features, their reliance on mutable state and sequential execution limits abstraction and composition.

In contrast, declarative programming languages–including *functional programming languages* such as Erlang, Haskell, and OCaml—are rooted in mathematics and lambda calculus (Rosser, 1941). They follow a declarative approach, constructing programmes from pure, referentially transparent functions that avoid side effects and mutable state. Features like immutability, higher-order functions, and recursion make functional programming well suited for concurrent and distributed computing, as pure functions are easier to parallelise and reason about.

Despite these advantages, functional programming can present challenges, including a steeper learning curve due to its abstract nature, potential inefficiencies in CPU and memory usage, and longer compile times due to advanced type checking and code generation. However, these trade-off's often lead to more robust, maintainable, and reliable software systems.

4.2.5 Using distributed data storage

Distributed computing is frequently used for data-intensive applications, which requires efficient data storage technologies to prevent persistence-related bottlenecks. While

traditional relational databases excel in structured data management due to their robust research foundation and optimisations, they often struggle to meet the scalability and flexibility demands of big data. To address these limitations, NoSQL databases have emerged, offering benefits such as horizontal scalability, schema flexibility, fault tolerance, and high availability –often at the cost of delayed consistency.

Unlike traditional databases evaluated on ACID properties (Atomicity, Consistency, Isolation, Durability), modern distributed databases are better evaluated using Brewer's CAP theorem (Brewer, 2012). This theorem states that distributed databases must prioritise either Consistency and Partition Tolerance (CP) or Availability and Partition Tolerance (AP). Highly available systems typically adopt eventual consistency to allow for non-blocking synchronisation. The main categories of NoSQL include key value stores, column-oriented storage, document stores, and graph databases, with open-source and proprietary implementations available (Siddiqa et al., 2017).

Beyond NoSQL, other big data storage solutions include parallel file systems and object storage systems. Although parallel file systems, such as Lustre (www.lustre.org), offer high performance in high-performance computing (HPC) environments, their strict adherence to POSIX standards limits scalability at very large data volumes. In contrast, object storage systems, such as Amazon S3, Apache Ozone, and MinIO (http://min.io), overcome these bottlenecks by using abstract data containers with immutable objects, stateless operations, and flexible metadata schemas (Liu et al., 2018). Initially suited for write-once, read-many workloads, improvements in performance, and latency have expanded their applicability.

In this study, the initial use of MongoDB (http://mongodb.com), a document-oriented NoSQL database, was chosen for its ability to handle large volumes of JSON documents, which matched the crop simulation input and output data format. However, the system was eventually switched to MinIO object storage due to its native compatibility with the Kubernetes environment, ease of deployment, and administrative simplicity. In practice, Poznan HPC cluster administrators found integrating MinIO to be more straightforward than setting up a network-connected MongoDB installation, highlighting the practical advantages of object storage solutions for scalable data management in distributed systems.

4.2.5.1 Simulation input data denormalisation

Traditionally, the required input data for the crop simulation model are stored and managed using a relational database management system (RDBMS), with a high degree of normalisation applied to minimise data redundancy and improve data integrity. As a result, however, collecting all data needed for a single crop simulation then requires a number of table joins or database views, making the data retrieval a relatively slow process. When we want to optimise computing the crop simulations, this retrieval of the input data

also has to be taken into account and optimised as well, or at least made suitable for the type of processing that needs to be done. Similar aspects apply to the processing and storage of the output data produced by the crop simulation model. Commonly, the data storage systems used for big data processing accept less data normalisation and actually prefer data duplication, where it serves to store the data on multiple computers so that it can be retrieved with higher parallelism by concurrently running processing jobs. Most of the NoSQL database systems described in Section 4.2.5 operate in such a way.

To transform the highly normalised data in the relational MARS database into a denormalised version, a Data Extractor programme was written that creates JSON Lines format output (http://jsonlines.org) with complete simulation input data per line of the file. An abbreviated single input record is shown in listing 4.1. For readability, it has been expanded across multiple lines; however, in the file it would span only one long line.

These files do get large (multiple gigabytes); however, with a simulation per line they are easy to process (e.g. filter, split, and merge) with standard operating system commands, and they can be significantly compressed for storage or exchange. Big data tools and distributed computing frameworks typically also have the capability to read compressed JSON files directly, although this might be time consuming depending on how well they manage to distribute the total workload.

```
{
  "version": 1, "simId": "
   grid1035126_crop2_variety20095_year1980_smu9030002_stu9000979",
  "simModel": "WOFOST", "simType": "waterlimited",
  "simCrop": 2, "simCropName": "GrainMaize", "simCropVariety":20095,
  "simYear": 1980, "simStartDate": "1980-04-28", "simEndDate":
   "1980-12-31",
  "location": { "type": "Point", "coordinates": [ 14.49247, 35.86522 ]
   },
  "sourceType": "CGMS",
  "sourceDetails": { "name": "Malta",
    "grid": 1035126, "smu": 9030002, "stu": 9000979, "altitudeM": 38,
    "gridWeightFactor": 0.111111
  },
  "cropParams": { "id": "crop2_variety20095", "params": [
    { "name": "SPA", "units": [ "[ha.kg-1]" ], "value": [ "0.0" ] },
    "..." ]
  },
  "soilParams": { "id": "stu9000979_smu9030002", "params": [
    { "name": "SMLIM", "units": [ "[cm]" ], "value": [ "0.3173" ] },
    "..." ]
  },
  "siteParams": { "id": "grid1035126_crop2_year1980_stu9000979",  "
   params": [
    {"name": "ANGSTB", "units": [ "[-]" ], "value": [ "0.42" ] },
    "..." ]
```

```
  },
  "agromanagement": { "id": "grid1035126_crop2_year1980", "params": [
    { "name": "IDEM", "units": [ "[date]" ], "value": [ "1980-04-28" ]
   },
    "..." ]
  },
  "meteo": { "version": 1, "id": "grid1035126",
        "startDate": "1980-04-28", "endDate": "1980-12-31",
    "data": {
      "temperatureMin": [ 11.3, 10.3, "...", 11.3 ],
      "..."
    }
  }
}
```

Listing 4.1: An abbreviated single sample JSON input record for simulation

4.2.5.2 Data serialisation and deserialisation

After extracting the data from the RDBMS in denormalised JSON Lines files, they can easily be imported into a MongoDB database (since it is based on working with JSON documents in collections) that can be accessed by Apache Spark, or Spark can read the JSON files directly for processing. In both cases, Spark handles the serialisation required and deserialisation from binary format with *Dataset Encoders*. To allow parallel operations in a cluster, Spark handles data via *Resilient Distributed Datasets* (RDD). RDDs are collections of (data) elements partitioned across the nodes of the cluster and that can be operated on in parallel. The *Dataset API* in Spark and the Encoder framework supports the construction of Datasets from JVM objects, and the manipulation of them using functional transformations (such as `map`, `flatMap`, `filter`, and so on). While Datasets are strictly typed, Spark also has a more generic *DataFrame API*, which is in essence a `Dataset` organised in named columns (a `Dataset[Row]`).

For this study, we had the advantage that we could define both the data schema and write the Java JVM objects that match it. By following the `JavaBeans` specification (i.e. make them serialisable, ensure a zero-argument constructor, and add accessor methods for all relevant properties), a standard Spark encoder factory (`Encoders.bean(...)`) could be used to create the Encoders from the `JavaBeans`.

The data schema we designed (for details, see 4.A) consists of a number of thematic blocks (crop, soil, site, agro-management, weather). Every block can have both a set of named parameters and a further flexible list of parameters. The named parameters are usually the key parameters of a block and can be used for grouping or sorting. In the future, these could be used, for example, to optimise the actual number of crop simulations to be performed by running only one simulation for groups that have exactly the same inputs. The flexible list provides the means of holding a variable number of additional parameters.

They are stored and retrieved, but are slightly more difficult and time-consuming to operate upon since they require unpacking first (Spark provides the functionality for this).

There is a schema for the input data (`SimulationInput`) shown in Table 4.A.1 and for the output data (`SimulationOutput`) shown in Table 4.A.4. In the input schema, `crop`, `soil`, `site` and `agromanagement` follow the same array-based schema containing `name`, `unit` and `value` structure per parameter (Table 4.A.3). Table 4.A.2 represents the schema of `meteo` providing weather data in time series between the start and end dates. On the other hand, in the output schema, `description` gives general information about the simulation (Table 4.A.6) and `message` contains a list of messages generated during simulation run (Table 4.A.4). In addition, `timeseries` provide the simulation results in time series (Table 4.A.7), while `summary` gives an overall summary of the simulation results (Table 4.A.5).

A small caveat, particularly when dealing with data that include geographic names (e.g., countries, regions, or cities), is to ensure that proper character encodings are used, such as `UTF-8`.

4.2.6 Using distributed computing

Distributed computing refers to the use of what is called a computer cluster with high available resources (mainly cores, memory, and disk space). The computational problem (the 'workload' or job) is typically divided into a number of tasks, and each of those is then solved by one or more computers. Or, in parallel computing jargon, each task is a sequence of instructions that operate together as a group. Tasks are mapped to Units of Execution (UE), which are the concurrently executing entities such as processes or threads. These UEs need to be further mapped to Processing Elements (PE), the actual hardware elements that execute the streams of instructions. The computers in a cluster are connected by a network so that they can communicate, exchange messages and data, and coordinate the work. When needed, there can be a final task that collects all the outputs and integrates or aggregates them to produce the end result.

Computer clusters can be built in various ways, mostly characterised by how memory is shared between computers and how instructions are executed on the data. Flynn's taxonomy (in Flynn, 1966 and Flynn, 1972) categorises the options in SISD (single instruction, single data), SIMD (single instruction, multiple data), MISD (multiple instruction, single data), and MIMD (multiple instruction, multiple data). The latter is the most well-known type, including computational grids, regular High Performance Computing (HPC) facilities, and Kubernetes. HPCs typically are tailored for high performance through the use of high-end components and allow computational jobs to be run by batch processing. That is, a job has to be entered into a queue and will be scheduled to run on the HPC based on the priority and availability of requested resources. Kubernetes environments usually allow more dynamic processing and are more flexible in adding and removing additional

resources based on demand. Today, it has become relatively easy to acquire computing resources from cloud providers to use temporarily and pay for them based on usage.

Distributed computing systems rely on middleware software to manage resources and tasks while abstracting low-level hardware details from users. In high-performance computing (HPC) environments, tools like SLURM (http://slurm.schedmd.com) and Torque (http://adaptivecomputing.com/cherry-services/torque-resource-manager/) serve as workload and job managers, often accompanied by modules for pre-configured software libraries or containerised applications, such as those packaged with Singularity (Kurtzer et al., 2017). In contrast, Kubernetes functions as both a platform and middleware, handling the scheduling of Docker (http://docker.io) containers as pods across available nodes and managing additional tasks, including those initiated by applications running within the system.

Modern open source software frameworks such as those mentioned in the Introduction section can help software engineers implement and deploy various kinds of data processing application on such systems. Their main goal is to improve the execution performance by reducing memory usage, disk I/O, and data shuffling based on optimisation and tuning techniques. Apache Spark, for example, uses the familiar split-apply-combine programming model and its specialised implementation called Map-Reduce (Dean and Ghemawat, 2008). This model has been heavily used internally by Google (until about 2015). MPI (an implementation of a Message Passing Interface) also uses partitioning and divide-and-conquer techniques to split the processing on the available resources and calculate partial results. While MPI is usually available as a programming library that allows low-level and optimised control over the way distributed processing is performed, the other mentioned frameworks offer medium- to high-level abstraction layers that provide a unified view of the available hardware and handle most of the resource allocation details during data processing. Typically, they support the development and testing of applications on a local computer with small sample data sets, after which the same source code can be deployed and used on a computing cluster to process the full data sets.

Since this use case revolves around performing batches of simulations with the Java based implementation of the WOFOST crop growth model, which will be described later, we choose to build on the Apache Spark framework, which uses Java Virtual Machines (JVM) internally as well making the integration easier. As a side note, Spark also supports the Python (PySpark), R (SparkR), and SQL (spark-sql) programming languages.

4.2.7 Experimental setup

For performance benchmarking, we will compare measurements with the current WOFOST implementation used in MCYFS, the European MARS Crop Yield Forecasting System that was introduced in Section 4.2.2. This system runs on Microsoft Windows 10.x, on a server with an Intel Xeon E5 CPU @ 2.3 GHz that has 20 cores in total and 128 GiB memory. In

comparison we run the system described in this paper on a physical GNU/Linux (kernel 5.x) server with also an Intel Xeon E5-v1 CPU @ 3.2 GHz, with 12 cores and 40 GiB memory. In addition, on a high-end laptop running macOS 12.x with an Intel Core i9 @ 2.4 GHz, 16 core and 16 GiB memory, and a small Kubernetes (v1.20.7) GNU/Linux (kernel 5.x) cluster with 1, 2, and 6 vCPUs @ 2.4 GHz, allocating 1 core (from an Intel Xeon E3-v2) and 3 GiB memory per vCPU (equal to a worker node in this case). This information is also included in Table 4.1.

Furthermore, the scalability of the system has also been benchmarked on a Kubernetes (v1.20.7) GNU/Linux (kernel 5.x) cluster, using larger numbers of bigger worker nodes with 4 cores and 16 GiB memory per vCPU @ 2.5 GHz (Intel Xeon E3-v2). The tests were carried out with 1, 2, 4, 8, and 16 worker nodes. Finally, on the Poznan HPC we ran scalability experiments with 32, 48 and 64 worker nodes running GNU/Linux (kernel 5.x) having 7 cores (Intel Xeon E5-v3) and 8 GiB memory each, using the high performance PSNC Eagle cluster.

In all cases Apache Spark version 3.0.2 and the same application (including the same WISS-WOFOST version) were used to execute the crop simulations and aggregate the results. OpenJDK 11 has been used as a JVM with the default garbage collector (GC) selected with no specific tuning done.

4.2.8 Distributing crop simulations with Spark

The Spark encoders described above can thus be used to convert the crop simulation input and output data between, e.g. a JSON representation and an instance of a corresponding Java JVM object. As mentioned, the WISS-WOFOST crop simulation model uses both a Java class that holds all input parameters, called `ParXChange`, and a class that collects all the output data produced by the simulation, the `SimXChange` class. WISS-WOFOST has been built so that all needed input can be passed to it via a `ParXChange` instance, which it can then use to run a crop simulation completely in memory. The output of the crop simulation is fully captured in a `SimXChange` instance. Via configuration, it can be set to write log messages e.g. to the console during processing. It is also built to fail fast when an error is detected, terminating with an exception. These can be captured for further handling.

Listing 4.B.1 in 4.B illustrates how only a few lines of code are needed to create the input (`SimulationInput` knows how to represent (part of) its data as a `ParXChange` instance) and output objects, an instance of the model, and use a `TimeDriver` to run a simulation. The `TimeDriver` object in this case externally drives the day-to-day time stepping of the model, until it reaches an end state. During the simulation, the calculated daily states (of the crop and the environment) are recorded in the output object. After completion, these states are used to update the `SimulationOutput` object. Finally, it should be noted that the type of result of the `run` method is a `Try[SimulationOutput]`, so it can be a success

with a `SimulationOutput`, or a failure with a Java exception (in Scala `Try` is a type that represents a computation that may either result in an exception, or return a successfully computed value).

The `object` (which in Scala defines a class that has exactly one instance, i.e. a singleton) and its `run` method from the listing 4.B.1 are further used as illustrated by the listing 4.B.2 (in 4.B) to perform the crop simulations in a distributed way using Apache Spark. Depending on the hardware on which it is used, Spark takes care of dividing the workload between the available cores and the computers, as illustrated in Figure 4.2.

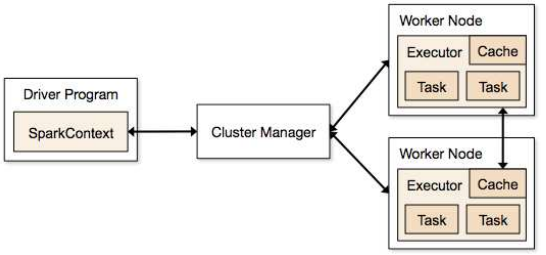


Figure 4.2: Main components of Apache Spark, including `Executors` deployed on Worker Nodes allocated from a computer cluster by the Cluster Manager. The Driver Program sets up the `SparkContext` used to control the distributed processing via `Tasks`. (From the Apache Spark documentation)

For that, the main programme (called the *driver program* of each Spark application gets a `SparkContext` object that handles the coordination of the processes of the application in a cluster. Via *cluster manager* (e.g. YARN or Kubernetes) which allocates the cluster's resources, it acquires *executors* on *worker nodes*. These are the processes that run the computations and store the data for the application. Next, Spark sends the application code (JAR or Python files) to *executors*, and finally sends them the *tasks* to perform. For their operation, *executors* needs to be able to communicate with each other and with *driver program*. Also, for performance reasons, it is best to run the *driver program* close to *worker nodes*, so in our study we always deployed it on the cluster as well instead of running it on a local computer.

Within a programme, the entry point to all Spark functionality is the `SparkSession` class. In listing 4.B.2 this is created at the beginning, followed by the construction of the `Encoder` instances for the `SimulationInput` and `SimulationOutput` classes to handle the input data deserialisation and output data serialisation (see Section 4.2.5.2). Since Spark is based on *lazy evaluation*, the next lines of code define the expressions to read the simulation input data from JSON file(s), perform crop simulation runs with WISS-WOFOST for all input splits into a requested number of partitions, and write the output of all successful simulations to JSON file(s). Note that it is the final `save` method that will require Spark

to actually evaluate all expressions and thus run the simulations. The code also calls the `cache` method on the `Dataset[SimulationOutput]` which holds the results of all crop simulations, so that it can be used for multiple types of (post)processing, including aggregation of the data as described in the next section. The schema of these data is shown in Tables 4.A.4, 4.A.5, 4.A.6, and 4.A.7 in 4.A.

The Spark `Dataset` (which is a typed version of the more generic Spark `DataFrame`) is a high level view of the partitioned data that represents it as a (very large) table with named columns (similar to a spreadsheet), while hiding all underlying complexity to the user. Typically, the amount of data to be processed makes it impossible to keep everything on a single computer and requires that it be distributed across multiple computers in the cluster. Still, for analytical purposes, Spark will make it appear as a single data set for which a processing pipeline can be specified in a declarative way, which it then translates into a number of possible *logical plans* for execution, of which the best is translated into a *physical execution plan* that is optimised for the available *worker nodes*. Unless it is needed to collect the content of a data set on a single computer (typically where *driver program* runs), it will keep it distributed and apply all requested processing accordingly.

4.2.9 Aggregating crop simulation results with Spark

The simulations with WOFOST are carried out for each unique combination of grid and soil unit in order to avoid aggregation errors caused by non-linearity in model responses. For defining these unique units, the European soil database is intersected with the simulation grid, leading to combinations of grid and so-called *Soil Mapping Units* (SMUs). However, the physical properties of the soil required for running the WOFOST soil water balance are not defined at the SMU level but are available for the so-called *Soil Typological Units* (STU). These STUs are not defined spatially, but only their percentage coverage of an SMU is known. Therefore, aggregating WOFOST simulation results requires one to take into account the interdependency of grid, SMU and STU. Moreover, final output is also required at the regional level, which involves aggregating from grids towards the lower administrative districts (the so-called *NUTS3 regions*) and to the national level (NUTS0). Furthermore, it should be noted that crop areas are not known at the level of grids or SMUs, only for the lowest administrative levels crop-specific area statistics are available.

The aggregation approach is visualised in Figure 4.3. At the lowest level, simulation results for each grid, SMU and STU combination can be aggregated toward the grid level based on a weighting factor computed from the configuration of STUs within the SMU and the area of each SMU with the grid. Then, aggregation was performed to the lowest administrative level taking the grid-level results and using the arable land area derived from a land cover map within each grid as a weight factor for the regional results. Finally, aggregation from the lowest administrative level towards higher levels was done by crop area statistics which are available from EUROSTAT.

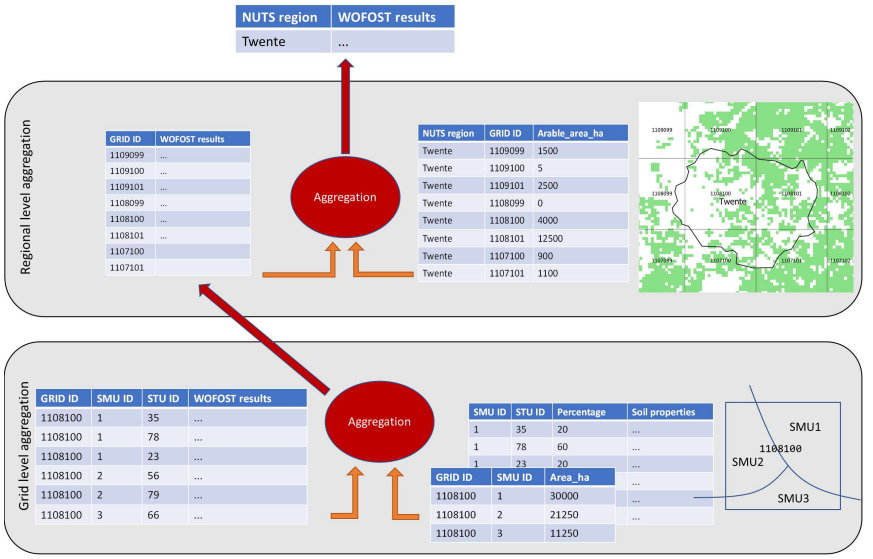


Figure 4.3: Schematic representation of the WOFOST result aggregation from individual crop simulations for unique combinations of inter-cell Soil Mapping Unit (SMU) and Soil Typological Unit (STU) to grid cell level (lower grey box) and from grid cells to regional level (upper grey box).

Despite the complexity of the aggregation scheme, the weight factors that are used to aggregate the results are static and can be computed a priori. Thus, they can be provided as an input to the simulation (e.g. in the JSON Lines formatted files) and replicated in the output from WOFOST. The aggregate of WOFOST results to the grid/region level can therefore be carried out with Spark by grouping on the grid/region ID, multiplying the WOFOST output by the respective weight factor, summing up the results, and writing the output to a new Spark `Dataset`. Aggregation of results from the lowest administrative level towards higher levels was not done from within Spark because the weight factors vary by year (e.g., annual crop area statistics). Moreover, the two aggregation steps already reduce the size to such an extent that adding this final step does not provide a substantial reduction of processing time.

Technically, aggregation is implemented in Spark using its select, group, and aggregation methods available for `Datasets`. It could also have been done by using its SQL API, but the total query needed is rather complex, and writing it in programming code makes it more manageable. The complete code is included as listing 4.B.3 in 4.B.

Finally, listing 4.2 shows a sample output record in JSON format, with the daily time series abbreviated. `TAGP` indicates the *Total Above Ground Production* (dead and living

plant organs) in kg/ha, and `TWSO` the *Total Dry Weight of Storage Organs* (dead and living) in kg/ha. More information about these variables can be found in the WOFOST system description (De Wit et al., 2020b). The values are the aggregated results of the water-limited crop simulations for maize, at the indicated grid location.


```
{
    "ERR": 0,
    "Total_Weights": 1.0,
    "NonErr_Weights": 1.0,
    "Crop_Code": 2,
    "Crop_Name": "GrainMaize",
    "Crop_Variety": 20102,
    "Grid": 1102097,
    "Region": "Zuid-Limburg",
    "Latitude": 50.96572,
    "Longitude": 5.60555,
    "Model": "WOFOST",
    "Simulation": "waterlimited",
    "Year": 2020,
    "Date": [ "2020-05-20", "2020-05-31", "..." , "2020-12-20",
   "2020-12-31" ],
    "TAGP_at_Date": [ 41.44378, 347.10842, "..." , 15950.85701,
   15950.85701 ],
    "TWSO_at_Date": [ 0.0, 0.0, "..." , 3821.31834, 3821.31834 ]
}
```


Listing 4.2: A single sample JSON output record after aggregation

4.2.10 Deploying on distributed computing infrastructures

Since Apache Spark can be used with various hardware configurations, it allows us to deploy and test the distributed crop simulations system on several setups. The simplest of them is a laptop, desktop computer, or single server. This can be used to develop the software and to run crop simulations on a limited scale (less than 1 million, depending on the hardware). Spark can use all available CPU cores for processing, so it can perform reasonably well on small amounts of crop simulations. It does, of course, introduce some overhead, so a single-computer setup is not necessarily the best approach in this case. In addition, it will not work for data sets that are too large to fit in the memory of the computer. This will require Spark to start overflowing data to temporary files, which reduces the overall performance.

To be able to process larger numbers (over 1 million) of crop simulations within acceptable time frames it is needed to use Spark with a configuration of multiple computers, such as a compute cluster. At the high end, these computers can be nodes of a supercomputer that provides High Performance Computing (HPC) facilities. Spark can be deployed on a subset of these nodes and then used to run the crop simulations. We have experimented

with such configurations by using our university's HPC and the larger facilities from the Poznan Supercomputing and Networking Centre (PSNC, http://www.psnc.pl). A supercomputer can provide many compute resources at relatively low cost, but they are very static and are usually operated in batch mode. Every processing job is entered into a queue and then has to wait its turn and until all requested resources are available. Running Spark on a Kubernetes (K8S) cluster provides a more dynamic approach but is typically also more expensive. The compute resources in a K8S cluster can be increased and decreased on demand, and Spark can use them directly to do additional work or more work at the same time. We have used Spark in a small K8S cluster provided by UBITECH (http://ubitech.eu) in various configurations to test processing scalability.

For deployment, the programming code to access input data, run crop simulations, and aggregate the large amount of simulation results needs to be combined into a Spark application. This needs to be packaged together with all dependencies (code libraries) it requires, so that Spark can distribute everything across all available worker nodes. A script, called `spark-submit`, is available to launch the application on a cluster. Depending on the `cluster manager` (e.g. SLURM or Kubernetes), this submit script can be called directly from the command line, or integrated into a cluster-specific deployment script. In the listing 4.C.1 (in 4.C) an example is given for a small-scale deployment (using 4 nodes) in our university HPC. In this case, SLURM will assign the requested nodes, and Spark will then automatically take care of setting them up as *worker nodes*, distribute the software (and data if needed) and start the application as *driver program*. While Spark handles most of the needed distribution and collection of data over the nodes, the log messages that might get written on each node are typically handled and automatically transferred to the user's home directory on the cluster by the cluster management software.

4.2.11 Summary

For this study, we took the standard and open source Apache Spark big data analytics framework and tested if it could be used to run large numbers of crop simulations with an existing version of the WOFOST crop growth model. Although Spark provides API's for Python and R, it is JVM-based at its core so we chose to use the Java WISS-WOFOST implementation to avoid performance loss due to data marshalling between programming environments and runtimes. All input data needed for the simulations were extracted from a relational database, denormalised, and stored in MongoDB or MinIO. These NoSQL types of storage better support distributed and parallel data access and can avoid I/O bottlenecks. Regular Spark SQL commands were used to aggregate the results of all individual crop simulations to grid cell and regional levels. The setup has been tested and benchmarked on a small Kubernetes cluster and large HPC configurations, which is further described in the following section.

4.3 Results

4.3.1 Overall performance

To benchmark the Spark-based WISS-WOFOST approach, we analysed the execution times of a test data set on different hardware configurations (see Section 4.2.7) and compared them with the execution time of the current WOFOST implementation used in MCYFS (the baseline). Table 4.1 summarises the different hardware specifications used. *Parallelisation* indicates the number of processing elements (see Section 4.2.6) used to perform the crop simulations (in parallel), while *Processing* lists the total execution time measured. Note that MCYFS did not fully use all available cores. The total workload (*Simulations* in the table) for the operational system by default consists of more than 3.7M simulations, while the test data set we used is much smaller, 113K simulations. This has been taken into account when computing the execution time per simulation ($Time/Sim_{raw}$). Moreover, the results for the MCYFS system only contain the execution time for *Processing* while for the Spark-based system the results cover both *Processing* and *JSON preparation* taking into account that I/O reads have to be done in both cases. That is, MCYFS retrieves the simulation data directly from its database, while in the other cases this data is first extracted from the database into JSON files, which are subsequently loaded into the system. Both steps are part of the indicated *JSON preparation* time, which is constant here due to Spark only using the *driver program* node for the read I/O in these experiments.

Table 4.1: Overview of various hardware configurations and measured crop simulations processing times. The $\texttt{Time/Sim}_{\texttt{raw}}$ times include both the processing time and the JSON preparation time when available. From that the $\texttt{Time/Sim}_{\texttt{scn}}$ single-core normalized execution time is calculated by inverse-scaling and assuming perfect parallelization.

Characteristic	MCYFS	Server	Laptop	Cluster 1	Cluster 2	Cluster 3
CPU type	Xeon E5	Xeon E5	Core i9	Xeon E3	Xeon E3	Xeon E3
Clock	2.3 GHz	3.2 GHz	2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz
Cores	20	12	16	1	2	6
Memory	128 GB	40 GB	16 GB	3 GB	3 GB	3 GB
Parallelisation	10	12	16	1	2	6
JSON preparation	N/A	18.2 min	18.2 min	18.2 min	18.2 min	18.2 min
Processing	4153 min	14.7 min	2.4 min	34.0 min	20.0 min	7.5 min
Simulations	3.773.853	113.662	113.662	113.662	113.662	113.662
$\texttt{Time/Sim}_{\texttt{raw}}$	0.0660 s	0.0174 s	0.0109 s	0.0276 s	0.0202 s	0.0136 s
$\texttt{Time/Sim}_{\texttt{scn}}$	0.6600 s	0.2088 s	0.1744 s	0.0276 s	0.0404 s	0.0816 s

The single-core normalised execution time ($Time/Sim_{scn}$) corrects for the number of processing elements that are used to execute the WOFOST simulations by inverse scaling (e.g., the total time would double when going from 2 cores to 1). Like the *raw* value it includes both the JSON preparation time when applicable and the crop simulation

processing time. For simplicity, perfect parallelisation is assumed in the calculation. In reality, this is never the case and actual *scn* values can be expected to be a little lower.

The results demonstrate that the Spark-based framework considerably reduces the processing time required for individual simulations compared to the baseline MCYFS system. Also, the use of more modern computer architectures still has a significant impact given that the Linux server with an older Intel Xeon E5 generation CPU is far slower (single-core normalised execution times) compared to the high-end laptop (Intel Core i9) and the clusters 1-3. Moving from single machines to multiple machines for the small clusters shows both the advantage of the overall processing times being significantly reduced due to the distribution of the workload, but also the cost of the increasing overhead by the higher $Time/Sim_{scn}$ values when more compute nodes are being introduced.

Some clarifying remarks apply to these benchmarks: (i) The measurements are only useful for relative comparison between the various configurations, the calculated single-core normalised execution times are indicative at best. (ii) JRC is working on a new MCYFS system that is based on a similar distributed approach based on the Python implementation of WOFOST (PCSE), which is said to show comparable computational behaviour as the WISS-WOFOST based implementation described here. (iii) The MCYFS processing measurements in the table have been calculated by analysing application log files of 10 parallel tasks that run crop simulations on the system, instead of active monitoring of a running system (which was no longer available at the time of writing). However, this should not affect the results. (iv) We did not compensate for the higher clock rate of the (Linux) server (3.2 GHz versus 2.3 - 2.4 GHz of the other configurations), since the net effects are hard to estimate and can only increase the gap in performance already shown.

4.3.2 System scaling

In addition to comparing performance, we also analysed the scalability of the system. Specifically, we looked at how adding more compute resources affected the total runtime required for processing two larger crop simulation input data sets. For this study, we selected all maize and winter wheat simulations between 2000 and 2020 from the MCYFS EU27 archive. The first data set has 3.8M (3 869 586) records and the second 5.1M (5 137 804) (each record contains all the data needed for a single-crop growth simulation). Both were used as input for running crop simulations on a Kubernetes cluster consisting of 1, 2, 4, 8, and 16 nodes (and the same number of Spark Executors) with 4 cores and 16 GiB per worker node. Table 4.2 shows the total processing times measured per experiment.

Table 4.2: Processing times for all maize and winter-wheat crop simulations between 2000-2020 from the MCYFS EU27 archive, on a Kubernetes cluster with various configurations.

		Processing time	
Spark Executors	**Total Cores**	**Maize (3.8M sims)**	**Winter-wheat (5.1M sims)**
1	4	670.9 min	820.9 min
2	8	324.9 min	445.9 min
4	16	164.1 min	219.2 min
8	32	84.3 min	137.8 min
16	64	48.4 min	88.2 min

The plot of the processing times on a graph (see Figure 4.4) clearly shows the relation with the number of worker nodes used and that after a certain number of nodes adding more resources will no longer be cost-effective.

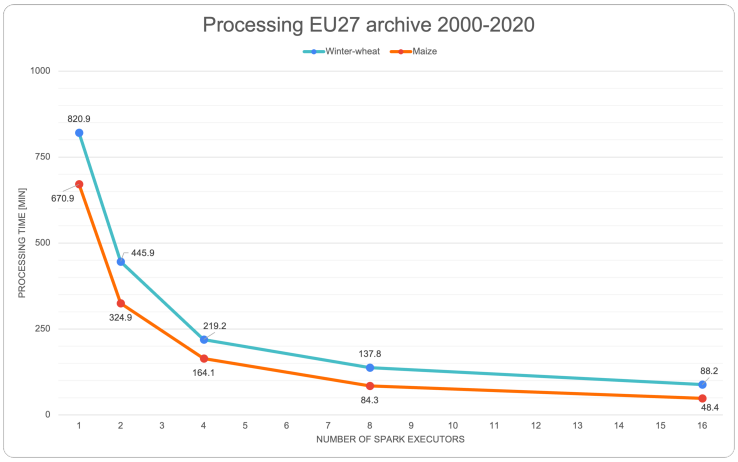


Figure 4.4: Plot of the processing times for all the maize and winter-wheat crop simulations between 2000-2020 from the MCYFS EU27 archive, on a Kubernetes cluster with various configurations. The diminishing benefit of adding more nodes is clearly visible.

The general scaling behaviour is similar for the 3.8M (maize) and 5.1M (winter-wheat) simulations. However, an important difference between these two data sets is that the latter, winter wheat, is a winter crop that has a significantly longer growing season (the period between sowing and harvest). So not only does the data set have more simulation records, each record also requires more storage space due to the longer time series of weather data it contains (the weather data are the largest component of a simulation record). Since Spark splits the total workload (all simulations to perform) into a specified number of partitions to be processed in parallel (using the cores available to each Spark Executor), both mentioned factors affect the memory space needed per partition, as well

as the JVM GC (garbage collection) (background) process that periodically frees up deallocated memory blocks. Besides that, depending on the processing workflow partitions sometimes have to be exchanged between the worker nodes (an operation known as a shuffle in Spark), which is a time consuming operation. These, and other considerations, make the number of partitions an important tuning parameter for Spark performance. Tables 4.3 and 4.4 illustrate this. On the same system, using 400 partitions is more performant for the maize simulations, while for winter-wheat it is 200 partitions (more optimal choices might exist). In these two cases, the effect is not very large. However, in an operational setting, such differences add up over time and are worth keeping in mind.

Table 4.3: Processing times for 3.8M maize simulations with different numbers of partitions, measured on a Kubernetes cluster with various configurations.

		Processing time (Maize, 3.8M sims)		
Spark Executors	**Total Cores**	**200 partitions**	**400 partitions**	**800 partitions**
1	4	670.9 min	**668.9 min**	681.7 min
2	8	324.9 min	**322.9 min**	330.2 min
4	16	164.1 min	**162.9 min**	183.2 min
8	32	84.3 min	**83.1 min**	98.5 min
16	64	48.4 min	**47.6 min**	56.5 min

Table 4.4: Processing times for 5.1M winter-wheat simulations with different numbers of partitions, measured on a Kubernetes cluster with various configurations.

		Processing time (Winter-Wheat, 5.1M sims)		
Spark Executors	**Total Cores**	**200 partitions**	**400 partitions**	**800 partitions**
1	4	**820.9 min**	827.4 min	836.0 min
2	8	**445.9 min**	461.3 min	461.3 min
4	16	**219.2 min**	228.6 min	239.7 min
8	32	**137.8 min**	144.7 min	164.8 min
16	64	**88.2 min**	94.4 min	104.6 min

The second study of system scalability was performed using the HPC of the PSNC Eagle cluster. This has a large number of nodes with 2x14 cores and 64-128-256 GiB memory, from which we used 32, 48 and 64 instances (with one Spark Executor each) and assigned with 7 cores and 8 GiB memory. In practice, the cluster manager could therefore combine two of these instances on a physical node or keep them separated; however, this should not affect the performance measurements. There is a clear, and to be expected, significant advantage of the HPC hardware (such as Infiniband networking with very high throughput and very low latency) over the Kubernetes Cloud hardware. Even when considering the increase in Spark Executors and the number of cores assigned per Executor. The measured execution times for the crop simulations are summarised in Table 4.5. The processing of the larger winter-wheat data set also allows Spark to use additional worker nodes and cores more than the smaller maize data set, where the advantages are marginal. However,

comparing the fastest processing of all maize simulations with the standard MCYFS system, we observe a significant 99% reduction in total runtime on hardware that has a 64-times higher distribution factor (64 nodes versus 1 node) and in total uses roughly 45 times more cores (448 cores versus 10 used by MCYFS).

Table 4.5: Maize and winter-wheat crop simulations processing times on a HPC cluster using various configurations with large numbers of cores.

		Processing time	
Spark Executors	**Total Cores**	**Maize (3.8M sims)**	**Winter-wheat (5.1M sims)**
32	224	5.43 min	13.81 min
48	336	5.20 min	10.25 min
64	448	4.79 min	7.55 min

4.3.3 Data aggregation

As discussed in Section 4.2.9 the total workload usually consists of two parts. The execution of a large number of crop simulations for grid cell / SMU / STU combinations, followed by aggregation of the simulation results first to grid cells and later to administrative regions for reporting. In the MCYFS an Oracle relational database is used to calculate the aggregated data, after first all simulation results have been ingested (and indexed). This loading of the data can take some time, but after that the needed database operations are typically fast. As an alternative option, we explored using Spark for the initial data aggregation (to grid cells). Due to the partitioning of the data on networked nodes, the required `groupBy` and similar Spark operations are known to be costly (in time). However, aggregating with Spark significantly reduces the amount of data needing to be imported into the database afterward, and it might not always be required to store the full detailed output of every crop growth simulation run (when necessary, they are fast to rerun).

Table 4.6 shows the measured processing times of the two data sets in the Kubernetes system. For this experiment, we only used the configurations with 8 and 16 nodes, again with 1 Spark Executor per node and 4 cores assigned to each executor. For simplicity, we estimate the time needed for the data aggregation by comparing the processing time of all simulations followed by the aggregation and the processing time of all simulations only. Regular SQL operations, such as used for the aggregation, are heavily optimised by Spark, and combined with the required data shuffling, a detailed analysis of the system in operation would be needed to get more precise numbers. We considered this unnecessary for the more superficial comparisons made here.

Table 4.6: Results for the maize and winter-wheat crop simulations followed by output aggregation from grid cell to regional level, on a Kubernetes cluster using 32 and 64 cores. Time used for the aggregation is estimated based on the measured total processing time.

		Configuration Spark Executors (cores used)	
Data set	**Result**	**8 (32)**	**16 (64)**
Maize (3.8M sims)	Overall processing time	89.24 min	50.44 min
	Only simulations	84.28 min	48.42 min
	Estimated aggregation time	4.96 min	2.02 min
Winter-wheat (5.1M sims)	Overall processing time	154.31 min	103.42 min
	Only simulations	137.82 min	88.16 min
	Estimated aggregation time	16.49 min	15.26 min

Comparing aggregation and data loading times between the MCYFS and Spark-based approaches is also complicated due to all the differences between the systems. However, to get at least an impression, we took the following approach. For both MCYFS (see Table 4.1 for the configuration) and Kubernetes deployment with 16 Spark Executors (using 64 cores total), we collected the processing times for crop simulations, data aggregation, and database loading, either by analysis of the application log files (in case of MCYFS) or by running the system (in case of Spark). To compare the aggregation performance in the database (using PL/SQL) and by Spark (Spark SQL), we estimated for both the number of STU records/sec and used this factor to calculate the estimated aggregation times for the two test data sets. Similarly, for database loading, we calculated the rows/sec loading speed and applied this factor to estimate the time it would take to load the Spark results into the Oracle database. The results are included in Table 4.7.

Table 4.7: Data aggregation and database loading times comparison between using the database for the aggregation (as done in MCYFS), and using Spark SQL to aggregate before inserting the results into a database (as done in our prototype system).

	MCYFS (agg in DB)		Spark (agg before DB)	
	Maize (3.8M)	**Winter-wheat (5.1M)**	**Maize (3.8M)**	**Winter-wheat (5.1M)**
DB Load	14.90 min	49.78 min	1.03 min	1.69 min
Data Agg	9.59 min	14.09 min	2.02 min	15.26 min
Total	24.49 min	63.87 min	3.05 min	16.96 min

We would like to note that with Spark, the data is aggregated before loading it into the database, while the MCYFS first loads all simulation results into the database and then performs the aggregation. In addition, the Spark configuration uses a significantly higher number of cores (64 versus 10 for the MCYFS), which helped to keep the aggregation time similar to the more optimised performance from the Oracle database. However, it clearly indicates that multiple aspects of a distributed system must be taken into account when evaluating its performance, as illustrated by Figure 4.5 (based on Table 4.7).

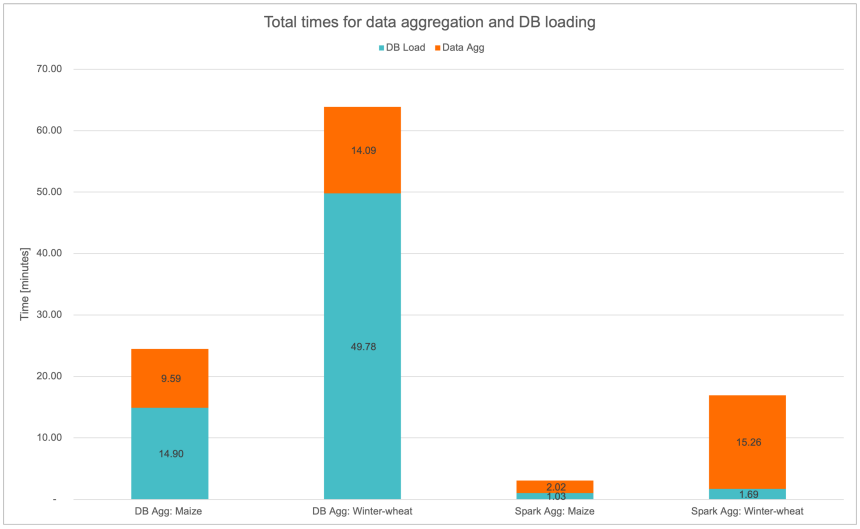


Figure 4.5: Combined times for data aggregation and database loading, when using the database to do the aggregation and when using Spark SQL. Results shown for both the maize and winter-wheat crop simulations.

As a final experiment, we used the PSNC HPC to run both crop growth simulations and data aggregation to the grid cell level with Spark SQL on large sets of nodes. For the results, see Table 4.8 and Figure 4.6. On the maize data set the benefits of adding more nodes are small, and the estimated aggregation time even gets slightly worse when the data is distributed across more Spark Executors. Still, comparing the estimated aggregation time on 64 nodes with 448 cores total, with that for the standard MCYFS system (1 node with 10 cores used) shows that a 95% reduction in processing time is achieved. Interestingly, on the larger winter-wheat data set, the medium configuration (on 48 nodes) shows less good performance improvements, most likely because the number of partitions is not ideal for the number of Spark Executors available. This illustrates once more the importance of performance tuning of Spark based systems in order to get the best results from them.

Table 4.8: Results for the maize and winter-wheat crop simulations followed by output aggregation on a HPC cluster using 224, 336 and 448 cores. Time used for the aggregation is estimated based on the measured total processing time.

Data set	Result	Configuration Spark Executors (cores used) 32 (224)	48 (336)	64 (448)
Maize (3.8M sims)	Overall processing time	7.32 min	6.11 min	6.02 min
	Only simulations	5.43 min	5.20 min	4.79 min
	Estimated aggregation time	1.89 min	0.91 min	1.23 min
Winter-wheat (5.1M sims)	Overall processing time	18.13 min	15.78 min	10.89 min
	Only simulations	13.81 min	10.25 min	7.55 min
	Estimated aggregation time	4.32 min	5.53 min	3.34 min

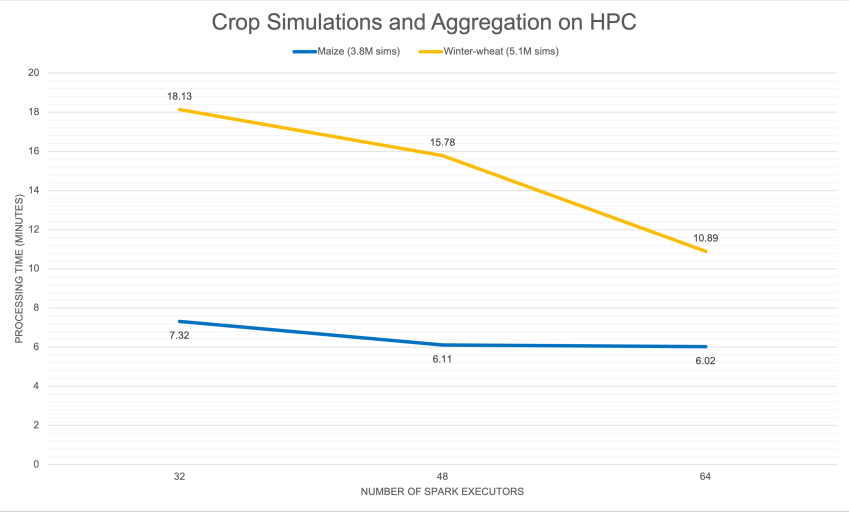


Figure 4.6: Plot of the results for the maize and winter-wheat crop simulations followed by output aggregation on a HPC cluster using 224, 336 and 448 cores.

4.3.4 Aggregation results

The aggregation process combines the output of the crop simulations first at the grid cell level (25 x 25 km) and then further at the lowest regional level. See Section 4.2.9 for more details. Taking the simulations of the maize crops as an example, the process results in 6.555 values for the Total Above Ground Production (`TAGP`) in dry weight kg/ha of biomass and for the Total Weight of Storage Organs (`TWSO`) in dry weight kg/ha, in 2020 for regions in EU27. The maps of these results are shown in Figures 4.D.1 and 4.D.2, in 4.D. Further aggregation can be performed for larger administrative regions.

4.4 Discussion

Using a typical numeric crop growth simulation model and a widely used standard framework for big data processing and analytics on distributed systems, we successfully built a prototype of the core of a yield forecasting and monitoring application. We then benchmarked its performance and scalability characteristics on a few different hardware configurations based on ephemeral resources such as those that can be used from cloud or HPC providers. The measurements were based on the processing of existing data sets for the maize and winter wheat crops and for a climate scenario. Although the data required for a single simulation is not that extensive, the large total number of simulations to be processed and the amount of output data produced place the task in the big data domain.

The obtained results illustrate that the expected effect of scaling on total run-time for performing crop growth simulations with the process model also applies, and thus allows trading-off compute costs with required result availability (within certain bounds). *Based on these findings we conclude that currently available standard distributed computing frameworks, specifically Apache Spark, are sufficient for building a system that is capable of running very large volumes of crop growth simulations.* Furthermore, the following insights can be deduced:

i. It is quite possible to base such a system on standard ephemeral compute resources managed by a commonly available open source framework, that hides a lot of the details of distributed computing, such as workload partitioning and hardware failure recovery, and makes it more easily accessible.

ii. A distributed system as described is capable of handling large amounts of crop simulations in reasonable time-frames, including all related (big) data management. The performed benchmarks provide insights into the performance and scalability that can be expected. As already common practice in other science domains such as climate sciences and hydrology, distributed technologies such as described in this paper are ready to be leveraged in the agricultural domain.

iii. Existing numerical models and data storage might need some adaptation before they can be *efficiently* used for distributed computing. In the case study little effort is needed to split a total workload into separate jobs that execute crop simulations simultaneously (in parallel computing this is referred to as an embarrassingly parallel workload). While an easy approach when applicable, it is not necessarily the most performant or power-efficient one. In this study little to no time has been spend on tuning the system to get the best possible performance. Spark has many configuration options that can be experimented with, the workload/simulations can be created or executed in a more efficient way, and the model (or parts of it) perhaps can be parallelised or replaced by faster solutions.

iv. A certain level of knowledge about the specifics of distributed computing and distributed data storage technologies will still be required, and comes with a learning curve that might not suite everyone. Specifically this will be needed when it comes to resolving issues (debugging the system) and for performance tuning.

v. In the case study we configure a prototype system for distributed processing of crop growth simulations using a specific implementation (the Java version) of WOFOST, and set up an *extract–transform–load* (ETL) pipeline to ingest the input data it needs for the crop simulations. Both on top of the generic Spark framework. Due the flexibility of the framework for communicating with models written in JVM languages, Python, and R, as well as to any command line oriented model (compiled or interpreted) via standard Linux pipes, it should be straightforward to incorporate any of the other WOFOST implementations available (De Wit et al., 2019). Other kinds of crop models will require additional configuration since the APIs and data requirements of these models can vary (Janssen et al., 2017; Porter et al., 2014). Still, it can be done as well, using the same framework and the methodology described in this paper.

vi. The prototype provides only the core of a potential operational system, as noted in Section 4.2.1, this was sufficient for the aim of this paper. As a follow-up the work can be extended into a production oriented application with e.g., a user interface that can hide some or most of the complexity from domain experts for day-to-day use.

vii. Spark is a generic distributed framework for batch processing and analysis of big data. Instead of the numerical model used in the prototype other kinds of models, including machine learning model training and inferences, can equally well be used. Alternatively, there are other frameworks available that specifically focus on machine learning (particularly useful for deep learning), or streaming data processing. Fortunately the basic concepts of distributing workloads across computers are the same for all these frameworks and experience with one easily applies to others.

viii. Various kinds of simulation output data post-processing and analytics can already be done on the same distributed system, which can reduce the size of the final results, making them easier manageable. An advantage is that the numerical model used is deterministic. Since individual simulations can easily and quickly be reproduced, there might be no need to store all output data from every simulation performed.

ix. The scalability of the system follows the expected curve with high gains on the initial increases of compute nodes, and decreasing benefits when the system gets bigger. Within limits, it allows a trade-off between acceptable compute costs and overall calculation time to obtain the result. Such flexibility is a benefit, but it also requires changes in thinking about costs of computations. E.g. they might no longer be fixed department IT costs, but rather become related to specific tasks or projects.

Although distributed technologies proved to be suitable for the case study and can similarly be expected to be equally applicable to other big data processing and analytics that are or will be part of digital agriculture, for example, for the processing of streams of data from many IoT devices (sensors) or for the computationally intensive 4D-Var data assimilation (Huang et al., 2019), they are yet not commonly known in the agricultural informatics domain, where most agricultural modellers and data scientists will first have to invest in the additional knowledge required. Naturally, the kind and depth of required expertise will vary, ranging from basic awareness of the existence and possibilities of the technology to expert and practical knowledge of distributed computing frameworks and, for example, functional programming (as mentioned in Section 4.2.4).

It can also be an important step going from personally curated data sets stored and processed on a local server to working with distributed, remote (perhaps even Cloud based), storage and processing of data. Some of the changes required are described in Section 4.2.5. However, in addition to those technology-oriented considerations, it can impact broader organisational aspects as well, since distributed computing and the sharing of resources such as infrastructure and data can go hand in hand. Furthermore, as a related aspect, users, managers, and developers will be confronted with new and different cost models, e.g., pay per use of temporarily allocated compute resources, which might not readily fit into the classical budgeting of IT resources.

Although the implemented prototype system described in this paper covers everything needed to create an operational version of it, there certainly remain a few interesting topics for future work:

i. In our approach, we take advantage of the WISS-WOFOST model implemented in Java, which matches well with Apache Spark, which is JVM-based. However, Apache Spark also supports Python, so a similar study could be performed using PySpark and the PCSE implementation of WOFOST (written in Python). However, this might incur a performance cost since data will need to be marshalled between the Python and the JVM environments.

ii. The described implementation is rather straightforward and takes an embarrassingly parallel approach for the computations. A possible optimisation is the minimising of the number of crop growth simulations (by sorting or caching results) that actually needs to be performed, e.g. by smart use of the named parameters we already included in the data scheme.

iii. As noted above, the weather data make up the largest part of each simulation input record. A possible improvement then is the broadcasting of these data to worker nodes and constraining crop simulation runs to those nodes that have the required weather data. This would reduce the size of the simulation record, overall I/O need, and thus increase performance and simulation throughput.

iv. Spark uses an optimisation technique called adoptive query execution, which uses run-time statistics to select the most efficient query execution plan. This works well for Spark SQL; however, the crop growth simulation process model is an opaque data transformation step in the processing pipeline to Spark, and therefore it cannot include it in its optimisations. If this can be improved, it should be beneficial for overall performance.

v. The data currently used includes (text) IDs for grid cells and administrative regions, which allow for simple text-based aggregations. Future data sets and aggregation approaches might not have this convenience and thus need support for geospatial data types and operations. There is quite a bit of history and development of spatial extensions on top of the Spark framework (both JVM and Python based), e.g. Apache Sedona (https://sedona.apache.org/) where many initiatives seem to end up. Integrating it into the system described in this paper would give it real large-scale spatial data processing and analysis capabilities.

vi. Although in this work a traditional numerical crop growth simulation model is used, deep learning and machine learning are increasingly prominent in agricultural data analysis and modelling, especially with large datasets. In addition to the comparison already suggested with the Python PCSE implementation of WOFOST, evaluations can be performed with machine learning-based models as the baseline, resulting in a broad overview of current technologies for high-efficiency agricultural modelling.

Note that while standard frameworks and cloud or HPC resources certainly make distributed computing more accessible and easier to use, building such an operational system is still not trivial. In particular, when it is the first encounter with building a distributed application, or with deploying such application on distributed cloud or HPC hardware. A framework such as Apache Spark can handle and hide many of the low-level details (data partitioning, task scheduling, handling batch and streaming data, hardware failure recovery, etc.), besides providing extensive data science and machine learning libraries for data analytics. However, ultimately it is still beneficial to know how the system is working and how it can be tuned to achieve the best possible performance, with acceptable costs. Fortunately, such knowledge is domain-agnostic and applicable to distributed systems in general, and thus can be left to specialised data engineers.

4.5 Conclusions

Taking into account the research objective, the case study described in this article and the benchmark results show that *a usable system can indeed be constructed on top of standard technology for big data analytics and distributed computing, in this case Apache Spark, and that it can provide a solution when true big data processing and analytics are required in the agricultural domain.* Generally speaking, the chosen case study is a regular big data

processing task, slightly complicated by the use of the numerical crop simulation model, which is not regular SQL that the framework knows how to fully optimise, but will remain a black box that can only be handled in a generic way. Still, the prototype leaves room for further optimisation and fine-tuning of the system. Furthermore, the applicability of the approach is not limited to this use case with a crop growth model. Due to the flexibility of the Spark framework for embedding e.g. Python and command-line based (compiled) models as well, it is a very generic solution.

In case data volume or computational requirements justify it, distributed computing using these standard technologies is a viable and powerful approach in digital agriculture. With the expected increase in demand for scalable processing–due the ongoing transformations towards data-driven agriculture–these technologies will become increasingly relevant. However, their use should be dictated by actual needs, due to the additional complexity involved.

It is important for agriculture stakeholders, agronomists, data scientists, and (spatial) data engineers, to understand distributed technologies, each in line with their role and expertise. Sensible application and cross-disciplinary collaboration will be the key to unlocking the full potential of distributed computing in agricultural informatics.

Acknowledgments

The work described in this document has been carried out as part of the CYBELE research project, which has been funded by the European Commission under the Horizon 2020 Programme (Grant Agreement No. 825355).

❄

Appendices

4.A Data schemas

Table 4.A.1: Overview of the schema for the crop simulation *input data*. Further tables show the details of subsections.

```
root
 |-- version: long
 |-- simId: string
 |-- simModel: string
 |-- simType: string
 |-- simCropId: long
 |-- simCropName: string
 |-- simCropVariety: long
 |-- simYear: long
 |-- simStartDate: string
 |-- simEndDate: string
 |-- location: struct
 |    |-- type: string
 |    |-- coordinates: array
 |    |    |-- element: double
 |-- sourceType: string
 |-- sourceDetails: struct
 |    |-- name: string
 |    |-- grid: long
 |    |-- smu: long
 |    |-- stu: long
 |    |-- altitudeM: long
 |    |-- gridWeightFactor: double
 |-- cropParams: struct
 |-- soilParams: struct
 |-- siteParams: struct
 |-- agromanagement: struct
 |-- meteo: struct
```

Table 4.A.2: Sub-schema for the meteo input data for a simulation. For efficiency the variables are stored as arrays which need to match the date range.

```
|-- meteo: struct
|    |-- version: long
|    |-- id: string
|    |-- startDate: string
|    |-- endDate: string
|    |-- details: struct
|    |    |-- grid: long
|    |-- data: struct
|    |    |-- temperatureMin: array
|    |    |    |-- element: double
|    |    |-- temperatureMax: array
|    |    |    |-- element: double
|    |    |-- temperatureAvg: array
|    |    |    |-- element: double
|    |    |-- vapourPressure: array
|    |    |    |-- element: double
|    |    |-- windSpeed10M: array
|    |    |    |-- element: double
|    |    |-- windSpeed2M: array
|    |    |    |-- element: double
|    |    |-- precipitation: array
|    |    |    |-- element: double
|    |    |-- e0: array
|    |    |    |-- element: double
|    |    |-- es0: array
|    |    |    |-- element: double
|    |    |-- et0: array
|    |    |    |-- element: double
|    |    |-- radiation: array
|    |    |    |-- element: double
```

Table 4.A.3: Sub-schema used for all crop, soil, site and agro-management parameters that are relevant but only need to be stored and passed to the model to run a simulation. For flexibility text string representations are used here.

```
|-- parameters: struct
|    |-- id: string
|    |-- params: array
|    |    |-- element: struct
|    |    |    |-- name: string
|    |    |    |-- unit: array
|    |    |    |    |-- element: string
|    |    |    |-- value: array
|    |    |    |    |-- element: string
```

Table 4.A.4: Overview of the schema for the crop simulation *output data*. Further tables show the details of the subsections. The `message` field is used to save any error messages that occur during the simulation, for later inspection.

```
root
|-- description: struct
|-- message: array
|    |-- element: string
|-- timeseries: struct
|-- summary: struct
```


Table 4.A.5: Sub-schema for the simulation output summary, consisting of the main WOFOST output variables, such as the Total Above Ground Production (`TAGP`) and the Total Weight of the Storage Organs (`TWSO`).

```
|-- summary: struct
|    |-- dvs: double
|    |-- laimax: double
|    |-- tagp: double
|    |-- twso: double
|    |-- ctrat: double
|    |-- cevst: double
|    |-- rd: double
|    |-- dos: long
|    |-- doe: long
|    |-- doa: long
|    |-- dom: long
|    |-- doh: long
|    |-- dov: long
```

Table 4.A.6: Sub-schema for the descriptive information about the crop simulation, indicating amongst others the location, the type of crop, and which crop growth simulation has been run.

```
|-- description: struct
|    |-- simId: string
|    |-- simModel: string
|    |-- simType: string
|    |-- simCrop: long
|    |-- simCropName: string
|    |-- simCropVariety: long
|    |-- simYear: long
|    |-- simStartDate: string
|    |-- simEndDate: string
|    |-- location: struct
|    |    |-- coordinates: array
|    |    |    |-- element: double
|    |    |-- type: string
|    |-- sourceType: string
|    |-- sourceDetails: struct
|    |    |-- altitudeM: long
|    |    |-- grid: long
|    |    |-- gridWeightFactor: double
|    |    |-- name: string
|    |    |-- smu: long
|    |    |-- stu: long
```

Table 4.A.7: Sub-schema with the time series data of key WOFOST variables during the simulation. These can be helpful to solve simulation issues.

```
|-- timeseries: struct
|    |-- date: array
|    |    |-- element: string
|    |-- elapsed: array
|    |    |-- element: integer
|    |-- dvs: array
|    |    |-- element: double
|    |-- lai: array
|    |    |-- element: double
|    |-- tagp: array
|    |    |-- element: double
|    |-- twso: array
|    |    |-- element: double
|    |-- ctrat: array
|    |    |-- element: double
|    |-- rd: array
|    |    |-- element: double
|    |-- sm: array
|    |    |-- element: double
|    |-- wwlow: array
|    |    |-- element: double
```

4.B Code listings

```
import nl.wur.json.{SimulationInput, SimulationOutput}
import nl.wur.wiss.core.{SimXChange, TimeDriver}
import nl.wur.wissmodels.wofost.WofostModel

object WofstRunner extends Serializable {
    import scala.collection.JavaConverters._

    def run(input: SimulationInput): Try[SimulationOutput] = Try {
        // get the input parameters and prepare an output instance
        val output = new SimXChange(input.getSimId)
        val result = new SimulationOutput()

        // fill a ParXChange instance from the input data
        val params = input.getParXChange

        // perform a daily timestep based simulation
        new TimeDriver(new WofostModel(params, output)).run()

        // collect input and output details into the result
        result.updateDescription(input, output)
        result.updateTimeSeriesSummary(input, params, output, true)
        result
    }
}
```

Listing 4.B.1: Code for embedding the WISS-WOFOST model in a Scala function

```
val spark = SparkSession.builder()
    .appName("WOFOST-Simulations")
    .getOrCreate()

// set up the encoders for the Dataset (row) types
implicit val inputEncoder: Encoder[SimulationInput] =
    Encoders.bean(classOf[SimulationInput])
implicit val outputEncoder: Encoder[SimulationOutput] =
    Encoders.bean(classOf[Simulationoutput])

// define the (JSON) input dataset
val inputData : Dataset[SimulationInput] = spark.read
    .option("encoding", "UTF-8")
    .format("json")
    .load(/* inputDataPath */)
    .as[SimulationInput]

// method to run simulations for all input records
def runByMapPartition(nPartitions): Dataset[SimulationOutput] = {
    inputData
        .repartition(nPartitions)
        .mapPartitions(_.map(WofostRunner.run))
}

// define how to produce wofost simulation results
val outputData = runByMapPartitions(/* partition count */)
    .cache()

// extract all successful simulations and save them
outputData
    .filter(array_contains(col("message"), "Ok"))
    .write
    .format("json")
    .mode(SaveMode.Overwrite)
    .save(/* failedSimulationsPath */)
```

Listing 4.B.2: Example of running WISS-WOFOST with Spark

```
// define how to produce wofost simulation output data
val outputData = runByMapPartitions().cache()

// define how to produce the aggregated result
val aggData = outputData
  // add a column with 0 if simulation succeeded and 1 if there were errors
  .withColumn("err", (!array_contains(col("message"), "Ok")).cast("integer"))
  .select(
    col("err"),
    col("description.simCrop").as("crop_code"), col("description.simCropName").as("crop_name"),
    col("description.simCropVariety").as("crop_variety"), col("description.simYear").as("year"),
    col("description.sourceDetails.grid").as("grid"), col("description.sourceDetails.name").as("name"),
    col("description.location.coordinates").as("coordinates"),
    col("description.simModel").as("sim_model"), col("description.simType").as("sim_type"),
    col("description.sourceDetails.gridWeightFactor").as("weight_factor"),
    // expression so that the sum of weight factors for successful simulations can be calculated
    expr("""case when err = 0 then description.sourceDetails.gridWeightFactor else 0.0 end""").as("non_err_weight"),
    // the timeseries are arrays, they require a bit more complex processing
    col("timeseries.elapsed").as("day"), col("timeseries.date").as("date"),
    expr("""transform(timeseries.tagp, x -> x * description.sourceDetails.gridWeightFactor)""").as("w_tagp"),
    expr("""transform(timeseries.twso, x -> x * description.sourceDetails.gridWeightFactor)""").as("w_twso")
  )
  .groupBy(
    col("crop_code"), col("crop_name"), col("crop_variety"), col("year"), col("grid"), col("name"),
    col("coordinates"), col("sim_model"), col("sim_type"), col("date"), col("day")
  )
  .agg(
    count("*").as("n_sims"),
    sum("err").as("sum_err"), // sum the number of simulations that had errors
    sum("weight_factor").as("sum_weights"), // sum all weight factors
    sum("non_err_weight").as("sum_non_err_weights"),
    // element wise summing of the values in the arrays
```

```
      elementwiseSumDoubleArrays(collect_list("w_tagp")).as("sum_w_
tagp"),
      elementwiseSumDoubleArrays(collect_list("w_twso")).as("sum_w_
twso")
    )
    .select(
      col("sum_err").as("ERR"), expr("round(sum_weights, 5)").as("
Total_Weights"),
      expr("round(sum_non_err_weights, 5)").as("NonErr_Weights"),
      col("crop_code").as("Crop_Code"), col("crop_name").as("Crop_Name
"),
      col("crop_variety").as("Crop_Variety"),
      col("grid").as("Grid"), col("name").as("Region"),
      col("coordinates").getItem(1).as("Latitude"), col("coordinates")
.getItem(0).as("Longitude"),
      col("sim_model").as("Model"), col("sim_type").as("Simulation"),
      col("year").as("Year"), col("date").as("Date"),
      // element wise processing again of the arrays
      expr("""transform(sum_w_tagp, x -> round(x / sum_weights, 5))"""
).as("TAGP_at_Date"),
      expr("""transform(sum_w_twso, x -> round(x / sum_weights, 5))"""
).as("TWSO_at_Date")
    )
    .orderBy(col("Crop_Code"), col("Grid"), col("Year"), col("Day"))

  // terminal action to perform the aggregation and save the result
  aggData.write.format(outputFormat).mode(SaveMode.Overwrite).save(
savePath)
```

Listing 4.B.3: Code for the data aggregation with Spark

4.C Deployment script

```
# -= slurm Anunna HPC =-
#SBATCH --time=02:00:00
#SBATCH --mem-per-cpu=4G
#SBATCH --nodes=4
#SBATCH --cpus-per-task=2
#SBATCH --job-name="wofost-runs"
#SBATCH --qos=std

module load spark/3.2.3-2.7
source $SPARK_HOME/wur/start-spark

# Spark config
spark_deploy_mode=client
spark_master=local[*]
spark_driver_memory=2g
spark_executor_memory=1g

# Set to proper path and file (needs to be absolute)
log4jconf=file:///home/WUR/[user]/wofost/wofost-spark-submit-v1/log4j.
    properties
# Folder with jars (can be relative)
jarsdir=./jars

# submit the spark job
spark-submit \
  --master ${spark_master} \
  --deploy-mode ${spark_deploy_mode} \
  --driver-memory ${spark_driver_memory} \
  --executor-memory ${spark_executor_memory} \
  --class nl.wur.json.WofostRun \
  --jars "${jarsdir}/WISSFramework-1.0.jar,${jarsdir}/WOFOST-WISS-7.2.
   jar,${jarsdir}/jafama.jar" \
  --conf "spark.eventLog.enabled=false" \
  --conf "spark.executor.extraJavaOptions=--XX:+PrintGCDetails --XX:+
   PrintGCTimeStamps" \
  --conf "spark.executor.extraJavaOptions=-Dlog4j.configuration=${
   log4jconf}" \
  --driver-java-options "-Dlog4j.configuration=${log4jconf}" \
  --files /lustre/scratch/WUR/[user]/wofost_inputs/[crop-simulation-
   input-data].json \
  ${jarsdir}/WOFOST-JSON-1.0-jar-with-dependencies.jar --url ${
   spark_master} \
  --mode json \
  --repartition 8 \
  --in [crop-simulation-input-data].json
```

Listing 4.C.1: Deploying a Spark application with spark-submit and SLURM

4.D Example output maps

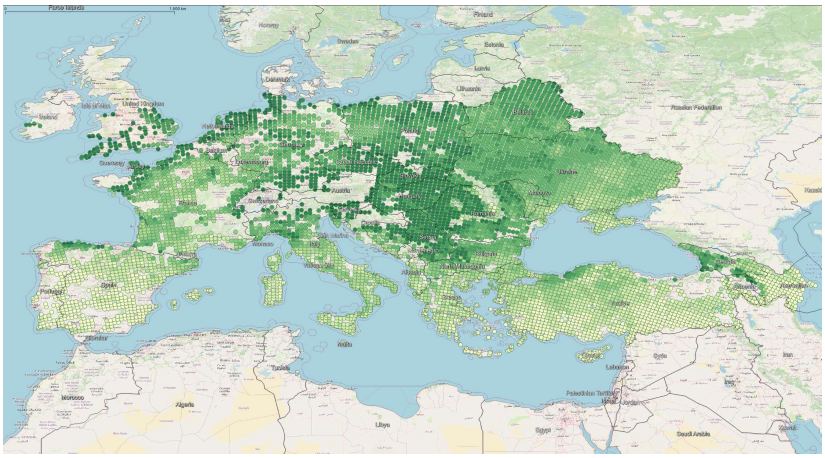


Figure 4.D.1: Map of aggregated Total Above Ground Production (TAGP) as dry weight in kg/ha from the 3.8M Maize crop simulations (2020, WOFOST water-limited). Showing lowest (700 kg/ha) to highest (29.000 kg/ha) values in light to dark shades of green.

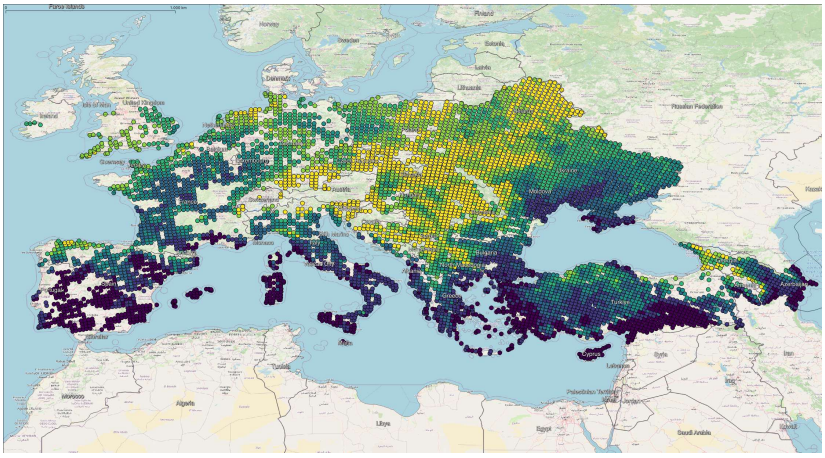


Figure 4.D.2: Map of aggregated Total Weight of Storage Organs (TWSO) as dry weight in kg/ha from the 3.8M Maize crop simulations (2020, WOFOST water-limited). Showing lowest (0 kg/ha) values in purple, via blue and green, to highest (14.600 kg/ha) values in yellow.

Chapter 5

Connecting Researchers

This chapter is based on:

M. Knapen, R. Lokers, and S. Janssen (2023). "Evaluating the D4Science virtual research environment platform for agro-climatic research". *Agricultural Systems* 210, 103706. DOI: https://doi.org/10.1016/j.agsy.2023.103706

Abstract

From its early beginnings the Internet has been used by scientists to collaborate and share information about their research. Increasing connectivity and networking capabilities have resulted in improved collaboration functionalities ultimately combined in complete virtual research environments (VRE) as a type of virtual laboratories. These aim at providing collaborative online workplaces with access to all needed tools, data, and computing resources, and supporting data sharing. Since each research domain has its own characteristics, requirements, and preferred tooling, VRE providers must make trade-offs between the specificity of components and the functionality provided. The D4Science VRE adopts a modular approach based on open standards for constructing VREs for interested communities. The agro-climatic science domain develops diverse analytical tools that it connects to heterogenous data sources (i.e. climate data, experimental fields, satellite data, soil samples) originating from other domains, which is often poorly standardised and sparsely interlinked at best. The aim of this paper is to test and evaluate the usefulness of the D4Science based VREs for this agro-climatic science domain, using crop growth simulation and crop phenology estimation as characteristic use cases, with specific attention to Open Science. Based on the needs of the use cases a VRE has been composed and further developed in an iterative approach and evaluated at the end of each implementation cycle. Both the development work and the evaluation results point at the foreseen potential benefits of adequate VREs and the current existence of sufficient opportunities and capabilities for constructing them. The focus when developing VREs should be on supporting research with proven and stable tools, instead of striving to include the latest and greatest. The agro-climatic research domain has ambitious requirements concerning the availability and integration of data and models, which proved to be particularly challenging for incorporating in a VRE. Yet, a clear but gradual adoption of digital techniques to further the science itself is happening and VREs represent an ultimate possible end-state of Open Science. To conclude, this paper provides a few recommendations that we think can help this ongoing transition.

5.1 Introduction

The agricultural system is a provisioning service that uses the natural environment to produce healthy and safe food. This agricultural system is facing a lot of challenges at the same time which are increasingly interconnected, for example, "How to feed the 9 Billion in 2050?" (Godfray et al., 2010). This challenge concerns not only the availability of sufficient food as a production and distribution challenge, but also the nutritional content of food and the nutritional diversity, as documented in a prominent definition of food security in 2001: Food security [is] a situation that exists when all people, at all times, have physical, social and economic access to sufficient, safe and nutritious food that meets their dietary needs and food preferences for an active and healthy life (Food and Agriculture Organization of the United Nations (FAO), 2002). Due to climate change, extreme events such as changes in the growing season, and changes in the production area directly affect the crop production and animal nutrition. At the same time, there is an increase in food-related diseases and health problems (i.e. obesity, malnutrition, cardio-vascular diseases, for example, (Micha et al., 2017)). Finally, all these challenges in the agricultural system must be achieved within our planetary boundaries (Steffen et al., 2015).

Given the interconnected nature of these challenges facing the food & agricultural system, a systems perspective adopted in science is a prerequisite to find smart solutions and to support decision making, either in policy or private sector organisations active in the agricultural system. Also, data about the agricultural system plays a crucial role to advance scientific discovery in new ways leading to a digital agriculture science (Janssen et al., 2017). There are a number of developments in agriculture and food science that support systems thinking. Firstly, an important development is the emergence of a system-perspective where the interconnectedness of challenges is highlighted, instead of being considered as separate independent issues, in the past referred to as integrated systems research (Harris, 2002; Jakeman and Letcher, 2003). This agricultural system perspective increases the complexity of the discussions on the short term, but in the long run ensures that responses can be much more targeted, balanced, and effective, while considering explicitly the trade-offs in the system and the unintended consequences of food system interventions.

Secondly, there is a rapid digitisation of the agricultural system, imprinted as a digital revolution, digital food, digital agriculture & big data (Janssen et al., 2017; Lokers et al., 2016a). Although this digitalisation has been promised, there are also still many open points requiring further developments, for example, the application of the FAIR data guidelines (Top et al., 2022), obstacles in data governance (Wolfert et al., 2017), and the applications of machine learning techniques and the ethical aspects involved (Amiri-Zarandi et al., 2022; Phillips et al., 2019; Rotz et al., 2019).

Third, a close collaboration with societal stakeholders in trans-disciplinary science or public-private partnerships leading to co- creation and co-design of the research has become an expectation for delivering more societal and impact relevant research (Cash et al., 2003).

Synergistic benefits are expected in agricultural systems research if trends in digitisation and systems thinking can be combined, grounded in the Open Science developments supported by developments such as FAIR data (Wilkinson et al., 2016) and Virtual Research Environments (Barker et al., 2019). In other words, in such Open Science developments, particularly when an agricultural system approach is adopted, there is a strong imperative of scientists working together on multi- & trans-disciplinary solutions, for using jointly developed data science solutions and interconnected (FAIR) data sets. In a way, scientists need to be able to work together in a virtual room, using the same resources and tools, without being in the same office. Such developments could potentially accelerate scientific discovery in the food, agricultural and environmental sciences (Zervas et al., 2018), allowing better contributions of science towards solving the agricultural systems challenges. Zervas et al., 2018, as part of their roadmap, identify a range of challenges concerning technical developments, community developments and governance. A particular problem in these specific sciences is the lack of sharing and collaboratively developing data, code, and data-analytical solutions, which is partly the starting point to overcome many other obstacles within the research communities and in the setup of collaborative infrastructures.

Over the past decades Virtual Research Environments (VREs) (Ahmed et al., 2018; Gadzhev et al., 2018; Zuiderwijk, 2017) have been proposed as such collaborative tools, and their prominence has increased with the advance of open science on the political and institutional agenda. A VRE can be considered as a set of applications, services and resources integrated by a standards-based, service-oriented framework, which will be populated by the research and ICT communities working in partnership (Allan, 2009). The objective of this paper is *to test and evaluate Virtual Research Environments as a potential collaboration tool with researchers from the agro-climatic domain, based on use cases from that field. The following sub-questions are addressed in this evaluation: (1) What are the requirements of agro-climatic researchers in their research tasks, specifically to support Open Science?; (2) What would be the required functionality for a VRE to be successful in fulfilling these requirements?; (3) How can agro-climatic researchers benefit from a VRE and what are the implications for their work?; (4) What would be required from researchers and research organisations in order to adopt a VRE?*

This paper first gives a brief background on the Big Data issues faced by digital agriculture, a brief timeline of VRE related developments, and current steps towards the Open Science Cloud in Section 5.2, followed by a description of the VRE platform used and the two use cases that were set up for the evaluation (Section 5.3). In Section 5.4 the applied evaluation

methods are explained, and the associated results and insights described, considering the research questions above. Based on the evaluation, Section 5.5 provides a discussion on the usefulness of VREs in agro-climatic research, and in Section 5.6 we present our final conclusions and recommendations for next steps of integrated digital and Open Science in agricultural, food, and environmental sciences.

5.2 Background

This section provides brief information on the specific issues resulting from the increasing amounts of data generated by the ongoing digitisation in the agricultural system. It presents e-research (the use of information technology to support existing and new forms of research), and Virtual Research Environments in particular, as potential helpful technology for addressing some of the issues and places it into the context of Open Science and FAIR data developments.

5.2.1 A brief timeline of VRE related developments

Before the year 2000, in the early days of the Internet, networked computers were only supporting parts of the research process, for example, by providing file transmission, electronic mail, and bulletin boards. Since 2000 the growth of the Internet has been exponential, and smartphones and cloud computing have seen increasing popularity from 2009 onwards. This also provided new opportunities for developing improved e-research platforms, among others, VREs with increased functionality (see Table 5.1). From 2000 until 2010 more complete and integrative VREs came into existence, but as bespoke or custom solutions. Allan, 2009 provides a good overview of the extensive work done on VRE development in this period, with a focus on projects funded by Jisc (www.jisc.ac.uk), a UK not-for-profit company providing, advising on, and funding research in shared digital infrastructure and services for education and research institutions. The nature of a VRE, supporting targeted research workflows, means that it can only realistically be described in terms of its intended capabilities rather than its precise component parts, since the latter will and should evolve over time, depending on contemporary standards, requirements, and technological progress (Allan, 2009).

Table 5.1: Evolution of VRE types and capabilities over time.

Timeframe	VRE types and capabilities
Before 2000	*Initial virtual collaboration foundations:*ARPANET, e-mail, bulletin boards
2000 - 2010	*Bespoke VREs:*Globus; Portals and Science Gateways
2010 - 2020	*Standardised and Commercial VREs:*D4Science; Cloud Platforms
After 2020	*Integrative VREs: Open APIs, FAIR data, shared semantics*

With the rise of Cloud Computing starting around 2010 there has been a split into efforts to build VREs based on open standards (on top of the typical Internet standards) and reusable components, such as the D4Science Data Infrastructure (since 2014), and efforts by commercial organisations to provide VREs (mostly as part of their Cloud platforms, e.g., Google Earth Engine, Google Collaboratory, Microsoft Planetary Computer, and IBM PAIRS Geoscope). Considering the direction for next generations of VREs (see also the findings by Calyam et al., 2021), there appears to be a clear drive to make them more integrative, i.e., allowing digital tools already in use by a scientific community to be effortlessly added, and data and algorithms to flow between components. Such developments are encouraged by today's movement of (data) science toward findable, accessible, interoperable, and reusable (FAIR) data (Wilkinson et al., 2016), shared semantics, and increased provisioning of web-based Application Programming Interfaces (APIs) to make data and algorithms easier accessible.

5.2.2 Toward Open Science cloud and FAIR data

In Europe, some of the earlier projects and organisations focused on the sharing of ICT infrastructure and research objects (including data sets and publications) in 2015 proposed the European Open Science Cloud for Research in a position paper (EUDAT et al., 2015). In general, Open Science is interpreted as "the movement that aims at more open and collaborative research practices in which publications, data, software, and other types of academic output are shared at the earliest possible stage and made available for reuse. With the expectation that it leads to increased scientific and societal impact" (www.nwo.nl/en/open-science). It has a close link with the publication of the FAIR guiding principles (Wilkinson et al., 2016). After several years of prototyping and implementation the European Open Science Cloud (EOSC) is now more well-defined and considered by the Council of the European Union as the science, research, and innovation data space, which will be further connected with other sectoral data spaces, such as the common European agriculture data space. As part of the new European data strategy (digital-strategy.ec.europe.eu), these data spaces are thought to bring together relevant data infrastructures and governance frameworks to facilitate data pooling and sharing. They are encouraged to use common technical infrastructure and building blocks, which must emerge from existing and new sectoral frameworks. While using European developments as an example, these movements are naturally not limited to this specific region.

5.3 Materials and methods

In the AGINFRA PLUS EU Horizon 2020 project (plus.aginfra.eu, Assante et al., 2020) the D4Science Data Infrastructure and VRE platform has been used to set up and evaluate a VRE that can handle agro-climatic research related scientific tasks (aginfra.d4science.org). Researchers can use this VRE as a Web portal to log in and get access to the various

configured tools and added data sources. Provided models and algorithms can be used and new ones implemented, combined into workflows, and used for data processing and analytics. Other functionality includes tools for semantic data enrichment and Jupyter Notebooks (jupyter.org) for exploratory data science work. The VRE has been constructed in increments, with evaluations by its users and intended audience in-between and at the conclusion of the work. The range of invited participants has been widened for each consecutive event, starting within the project, and broadening via related networks and communities of agro-climatic researchers.

5.3.1 The D4Science platform

The foundation for the VRE implementations supporting the AGINFRA PLUS use cases is the D4Science platform, which adopts the system-of-systems approach (Maier, 1998) to offer a comprehensive platform for setting up and operating VREs with the as-a-Service delivery model. D4Science serves as a generic platform for the composition and deployment of Virtual Research Environments, offering the core services required for e-research, and allowing the setup and operation of dedicated, research domain-specific VRE instances as-a-Service.

The D4Science infrastructure and the associated gCube (www.gcube-system.org) open software stack offer a way to easily construct VREs by combining common and domain-specific components and services based on the use of open standards. It allows tailoring VREs towards community-specific workflows, although this can still require community-level implementation of specific interfaces before it can be executed.

D4Science uses open standards at several levels. Overall, it is based on familiar web-based standards (from W3C) and technology such as web services, portlets, and a message bus for inter-component communication. Basically, any web service that has a web front-end can be plugged in. At a deeper level, it uses the gCube open-source framework that is Java and Java Virtual Machine (JVM) orientated. It offers application programming interfaces (APIs) that can be used to write components that are tighter integrated into the core system and make use of small runtimes that provide the main gCube services. The standards used at this level will be mostly known to software developers, for example Google Web Toolkit (GWT) and the Java API for XML Web Services (JAX-WS).

The gCube framework furthermore provides a catalogue mechanism (gCat) that uses CKAN (ckan.org) as underlying technology, which is a well-known open-source data management system. gCat has a REST (representational state transfer (Fielding, 2000)) API that gives access to the catalogues, both the global one and the VRE specific ones, so they can be harvested by other catalogues. Similarly, items (metadata about scientific papers and data sets) can be collected from existing community-specific and general catalogues such as OpenAIRE (www.openaire.eu). It is also based on the DCAT (data catalogue vocabulary) and Dublin Core semantic metadata standards, like used e.g. by FAIR Data

Points (fairdatapoint.org) (although gCat does not fully implement such data points at the moment). Furthermore, tools such as OpenRefine (openrefine.org) and the Silk Workbench (silkframework.org) can be added to a VRE to help with semantically annotating research objects to improve their interoperability. Relevant terms from existing ontologies and vocabularies can be added for this (Assante et al., 2019).

The resulting platform is depicted in Figure 5.1, described in more detail in Assante et al., 2020 and made available through a dedicated AGINFRAPlus Gateway, described in the next section.

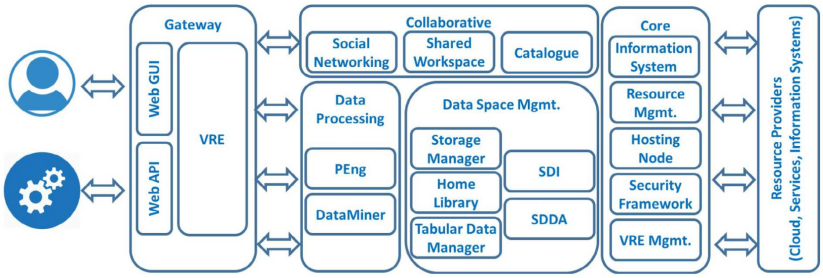


Figure 5.1: The complete AGINFRA PLUS platform architecture, with blocks indicating the various components included.

5.3.2 VREs for agro-climatic modelling

For the VREs developed under AGINFRA PLUS, D4Science offers (i) the AGINFRAPlus Gateway as a single access point to the VREs; (ii) services for authentication and authorisation, offering seamless access to shared services and components; (iii) a shared workspace, allowing users to store and share research artefacts like data sets, implementations of algorithms or models; (iv) a social networking area, allowing to share and disseminate information among VRE members, and facilitating discussions; (v) a catalogue, for publishing and sharing documented research assets with the community and the outside world. Moreover, D4Science offers a variety of standard components to support the research process, e.g., for semantically driven data management, data visualisation, and data analytics.

Where each VRE already brings various services together for end-users, the data analytics components provide researchers with access to distribute computing facilities as well. The two main tools offered for this are the well-known Galaxy workflows (galaxyproject.org) and the gCube DataMiner service (Assante et al., 2019). DataMiner provides simple procedures that allow native software, e.g., algorithms such as R or Python scripts, to be distributed on the available computers in a cluster and processed in a Map-Reduce fashion (Dean and Ghemawat, 2008). D4Science builds the cluster from available compute resources using

familiar web services standards, such as the Web Processing Service (WPS), the Simple Object Access Protocol (SOAP) and the Extended Markup Language (XML).

5.3.3 Use cases

The work in AGINFRA PLUS was organised around three scientific user communities working on the topics of agro-climatic modelling, food security and food safety respectively. These communities have defined typical research use cases relevant for their domain, which they expected would benefit from the capabilities of a VRE (Lokers et al., 2016a). These use cases represent a range of domain-specific research workflows, covering the whole chain of data acquisition, data processing, analysis, and publication. In current research practice, these workflows are usually supported by applying different methods and tools, largely depending on the detailed needs of the use case and often also on the personal preferences of a researcher and the access she and her organisation have to specific tools. In the following, we focus on the two use cases established specifically for the agro-climatic user community, and the requirements and types of functions and tools that are needed to perform the processes defined as part of these use cases. The first use case deals with large-scale agro-climatic modelling, with as main objective to automate and efficiently run large numbers of crop growth simulations using climate data and a variety of agronomic data. Followed by aggregation, visualisation, and publishing of the simulation results. The second use case focusses on crop phenology estimation, where the objective is to develop, run and share algorithms to derive crop phenology from remote sensing data using collaborative modelling to support the co-development and testing of algorithms.

While there are many implementation specific details, the use cases also share some common functional requirements, which are: (i) to acquire and be able to easily combine heterogeneous and often spatio-temporal large data sets and prepare them as input for algorithms; (ii) to develop models for distributed computing and to deploy and run them on compute clusters; (iii) to co-develop and test algorithms, using shared literate programming tools that allow a collaborative and agile approach; and (iv) to publish and share research assets for reuse in a standardised way, and to make use of assets shared by other communities. It is notable that the first two requirements refer to known data challenges of the agro-climatic domain. They are associated both with the massive sets of climate and remote sensing data, and with the heterogeneity of data (besides climate and remote sensing data also data about crops, soils, soil moisture and irrigation, agricultural parcels and many more). They also reflect the complexity of processing such data, dealing with the specifics of spatio-temporal data, and running models at massive scale on fine grained resolutions. The other requirements are typically associated with the concept of Open Science and the capacity to co-create and collaborate on research workflows and to be able to publish and share research assets with other communities and likewise to be able to access and reuse data, algorithms and other components developed by other research groups.

5.3.3.1 Crop growth simulation

Computer simulations based on crop growth models are one of the important components in yield monitoring and forecasting, used frequently in food security research and related research areas. Currently, the application of crop growth models is often still limited in scale and level of detail by the available computing resources. The application piloted in AGINFRA PLUS, applying European or world scale crop simulations at the detailed level of agricultural parcels is today considered too demanding for many existing research infrastructures (laptops, desktop computers, very small bespoke compute clusters) used in the domain. This can easily be illustrated by considering running simulations over the territory of the EU, with an estimated number of 100 million agricultural fields. Even if a crop growth simulation for a single field and 1 growing season takes only a few microseconds, say 10 milliseconds, calculating all simulations in sequence on a single CPU would take roughly 12 days, while requiring substantial data sets of crops, soils, and daily local weather, and producing simulation results per field for analysis. Therefore, to meet the requirements for such large-scale, high-resolution crop growth modelling exercises, the following facilities are indispensable: efficient retrieval of spatio-temporal data, spatio-temporal data wrangling and data processing, running models at scale using parallel computing and compute cluster resources, and intuitive spatio-temporal data visualisation.

In the AGINFRA PLUS crop growth simulation use case, the pre-processing of spatio-temporal data is an integral part of the AgroDataCube (Janssen et al., 2018) infrastructure (agrodatacube.wur.nl). This infrastructure provides Dutch agricultural open data as a service to research and business. The AgroDataCube ingests and harmonises different spatio-temporal data sets that are relevant for agricultural and environmental applications (among others, weather data, agronomic data, parcel geometries, Sentinel-2 satellite data, and soil data). It provides a set of well-documented, ready to use Representational State Transfer (REST, Fielding, 2000) services as Application Programming Interface (API) that allows retrieval of the merged data at the parcel level in usable packages and a standardised format (GeoJSON, geojson.org). To cope with the requirement of scaling up crop growth simulations, the widely used WOFOST crop growth simulation model (De Wit et al., 2019) was embedded into an actor-based framework for parallel computing (akka.io) and integrated with the Web Processing Service (WPS) based distributed processing provided by the D4Science software stack (Figure 5.2).

The resulting modules have been integrated into the VRE as DataMiner algorithms, and were published, using the D4Science catalogue service, to make them discoverable and reusable by the whole community. As the requirements for spatio-temporal data visualisation in this use case were substantial, a dedicated visualisation dashboard (Figure 5.3), developed based on various VRE components, has been added as well.

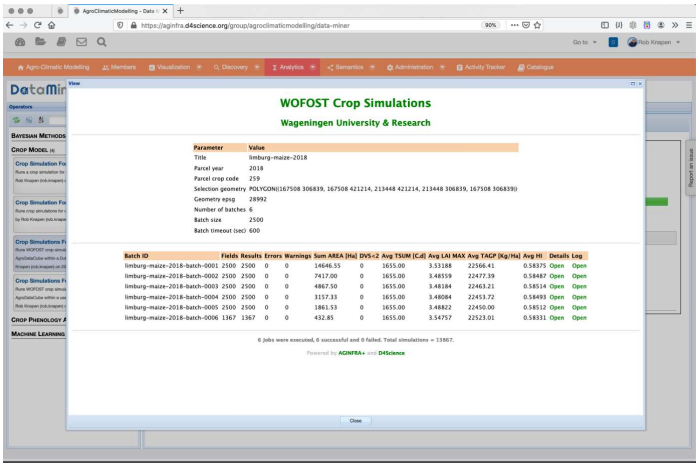


Figure 5.2: Logs of WOFOST computing jobs using the AGINFRA PLUS compute cluster.

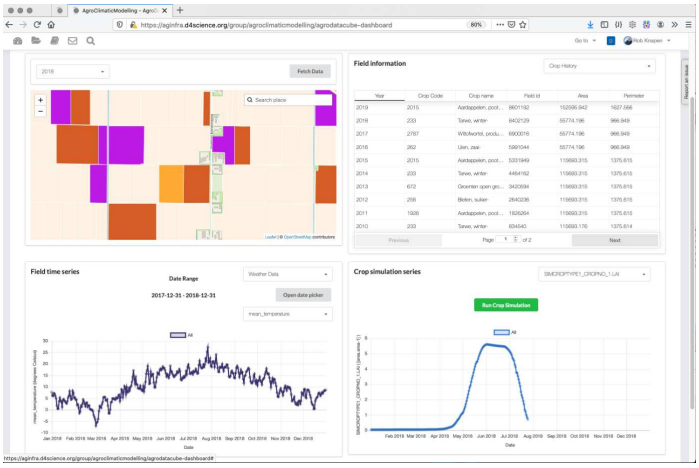


Figure 5.3: AGINFRA PLUS visualisation dashboard showing simulation results of individual farm parcels.

For visual inspections, this dashboard allows geospatial and temporal visualisation of the various data services provided through the AgroDataCube API that are input to the crop growth simulations. Moreover, it offers its users the opportunity to manually search for and select a specific agricultural parcel and initiate a WOFOST crop growth simulation by executing the VRE DataMiner algorithm using input data based on the selected field. All calculated simulation results are stored on the VRE's shared workspace, and can be visualised as graphs, as well as compared and analysed side by side with the used input data. After being tested and quality checked, the developed algorithms and the generated output data can be shared with the broader user community, by publishing them through the VRE's catalogue service. Thus, the VRE is complying with the requirements of FAIR (Wilkinson et al., 2016) data services and open science in general, adding to that the opportunity to also share algorithms in a FAIR-like, reusable manner. For more details see Knapen et al., 2020.

5.3.3.2 Crop phenology estimation

The development of an agricultural crop throughout the growing season, and the final amount and quality of harvestable produce depends on various factors. While climatic conditions are a major driver, soil conditions and farm management (e.g. applying pesticides, irrigation, soil management) also have a substantial influence. Being able to monitor crop development is a good means to support farm management and to be able to predict yields. Crop phenology, defining the physiological development stages of crop growth from planting to harvest, provides important information regarding the development of crops over time. Being able to assess crop phenology "in the field" allows researchers to verify the accuracy of crop growth simulations over the growing season. One possible way to estimate crop phenology is by analysing remote sensing images acquired by satellites. Satellite images can be processed to express the "greenness" of the earth surface and represent a measure for the development stage of a crop in a field. Various algorithms to calculate such vegetation indices exist, of which the Normalised Difference Vegetation Index (NDVI, www.indexdatabase.de/db/i-single.php?id=59) is generally considered to give a good indication of the development stage of a crop. The development of crops over time follows a relatively well-defined growth curve (see e.g. Vasilieva and Cherepanov, 2017), and the relation between the greenness curve and essential crop development stages has been extensively studied. The idea here is that by fitting a curve through the calculated NDVI values, it is possible to derive the crop development stages based on the NDVI (illustrated in Figure 5.4).

This use case has a direct link to the previously described crop growth simulation use case, as crop growth models heavily rely on crop phenology as a driving factor for the modelled processes. In principle, such information also provides the opportunity to adjust the model and improve model outcomes, e.g. through data assimilation of crop phenology estimates

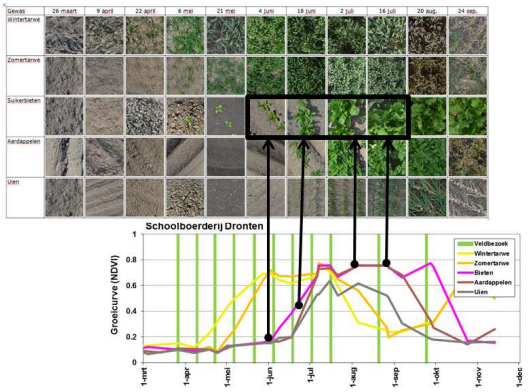


Figure 5.4: Relating crop development stages to NDVI. The small images show different types of crops in fields at selected dates in the Dutch growing season, taken at an educational farm. These dates match with the green vertical lines in the chart, which also shows growth curves based on NDVI values for the fields, derived from remote sensing observations. The annotations in black indicate the correlation between the NDVI curve and the actual phenological development of sugar beets.

and using them to adjust model algorithm parameters in order to close the potential gap between simulated results and observed reality.

There are several challenging aspects to this use case. First, satellite images are only available according to a specific acquisition plan (about every 2—3 days for Sentinel 2 at mid-latitudes) and are often only partly usable because of cloud coverage (for the Netherlands this seriously limits the available data, particularly in the autumn and winter seasons). Second, there are many factors that influence the crop growth process. The type of crop and variety itself is essential, but also other local factors like soil, climate, and farm management are relevant. Being able to quickly and efficiently develop, test, and improve algorithms helps in modelling the many parameters and complex interactions. The main objectives of this use case were to accommodate such a collaborative and iterative data science (Cao, 2018) approach for crop phenology estimation, to scale up the computations from farm field to a regional level, and to publish the outcomes. This includes experimenting with data analytics, working iteratively, combining different data sources, and making extensive use of data visualisation, while being able to collaborate and discuss results and improvements.

The development of the crop phenology estimation algorithms as part of the use case was performed using a literate programming (Knuth, 1992) approach, that mixes explanation of logic in natural language with snippets of computer source code. The D4Science platform hosts Jupyter Lab (www.jupyter.org) and RStudio (www.rstudio.com) as contemporary

components supporting this programming paradigm. For this use case, the Python (www.python.org) programming language and Jupyter Notebooks were used to facilitate the explorative modelling approach, applying the literate programming concept to facilitate co-development, and exploiting the vast array of available Python libraries for data management, (geo-temporal) data analytics and visualisation. Algorithms tested on the crop parcel level were eventually deployed as DataMiner algorithms and integrated in workflows to scale up from the parcel level to the regional level (the results are shown in Figure 5.5), exploiting the VREs facilities and resources for distributed computing. As in the crop growth modelling use case, the VREs publishing function and catalogue were used to share the final algorithm and make it available in a FAIR manner.

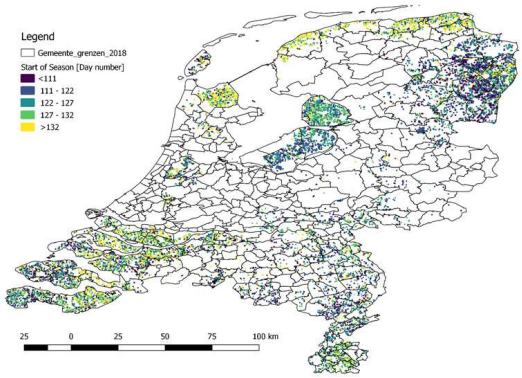


Figure 5.5: Example output for large scale "start of season" crop stage calculation performed on the AGINFRA PLUS Platform.

5.3.4 User evaluation

Even within a specific research domain like agro-climatic modelling, there are many different research approaches, and researchers and research communities have dedicated methods and preferred tools to perform their research. To ensure that such different approaches towards data science are accounted for when developing a virtual research environment, AGINFRA PLUS implemented an iterative approach of learning and improving as part of the development of its virtual research environments. For that reason, the e-research pilots with different user communities were set up as a three-phase trajectory, where each of the three development phases was concluded with a pilot evaluation performed by potential users of the system. The main aim of these evaluations was to allow researchers from different parts of the agro-climatic research community to work with the developed VRE components and workflows and to then survey them in a structured way on their experiences to discover the system's strengths and weaknesses. In this way, valuable data about the perception of independent users with different backgrounds could be collected

and analysed, and subsequently used to improve both the VRE platform and the developed components in the next development phase.

At the end of each of the three development phases, intensive pilot evaluations were conducted with a targeted group of researchers and data scientists to gather user experiences and perceptions around core VRE aspects like ease of use, learning curve and FAIRness. The groups were selected and invited based on their expertise in the use case domain and their involvement in data science in general. It must be noted that the characteristics of the user community differ per use case, and that different perspectives on the meaning and relative importance of evaluation topics might have influenced the way individual users approached the evaluation. However, this aligns with the main objective of these evaluations to identify strengths and shortcomings from the usage and user community perspective and to iteratively improve pilot functionalities.

All groups received a full day training and evaluation programme. First, they received an introduction to the concepts of e-research and were demonstrated the basics of the VRE and its components. Subsequently, participants independently worked on a set of research exercises, using the VRE components and workflows developed by AGINFRA PLUS. This setup allowed participants to not only learn, but to also perform hands-on work, and to relate it to their common scientific practice. After the training and evaluation exercises, participants were asked to report on their experiences and provide comments through a survey, which resulted in useful feedback that was used for improvements in the subsequent development phase.

After the last VRE development iteration, an additional validation exercise was performed with a larger group of researchers. In this case, the objective was not to collect data and inputs for further improvements, but to validate the results with a wider audience and create broader awareness and adoption. It particularly allowed to collect more feedback about how researchers and data scientists perceive the usefulness of VREs for research and the potential for their deployment in research organisations. A broader group of potential validators was invited, within and outside of the original application domains of the use cases. The focus of this validation was specifically aimed at providing all participants the same background information and providing a uniform survey to acquire harmonised, comparable results. This validation was performed through an online webinar, where participants learned about the VRE characteristic and received short demonstrations of agro-climatic modelling VRE applications. Again, participants were requested to subsequently participate in a survey, but this time focussing on assessing their perception of the usability of the VRE and to explore the opportunities and barriers they saw for uptake of a VRE in their work.

As part of the implementation work in the AGINFRA PLUS project, a group of researchers and data scientists was intensively involved in the configuration of the VRE, the setup of the research workflows in the described use cases, the evaluations and the formulation and

implementation of requests for change based on evaluation feedback. Their experiences with the D4Science platform and the development of VREs transcend the perspectives of external evaluators. Therefore, their most relevant insights and experiences are also shortly described.

5.4 Results

In this section an overview of the evaluation and validation results is presented. The full details are available as one of the AGINFRA PLUS project deliverables (Lokers and Knapen, 2018).

The AGINFRA PLUS evaluation surveys were conducted at the end of a full day VRE evaluation session. They were set up as semi-qualitative surveys, collecting scores and perceptions over 5 topics that are relevant for virtual research environments adopting the principles of Open Science: (1) Ease of Use; (2) Usefulness; (3) Openness; (4) FAIRness; and (5) Learning Curve. In total 22 of the participating evaluators have responded to the evaluation survey.

Regarding usefulness and ease of use of the provided VRE, many participants appreciated the overall VRE concept of being able to communicate, collaborate and co-develop in a "safe and controlled" remote environment (Figure 5.6). While some participants have also noted that although good tools supporting aspects of collaboration, analytics, and visualisation are already on the market, the bundling in one integrated and shared environment is generally acknowledged as an added value. Evaluators in general appreciated the options to collaborate, but we also observed that especially data analysts and model/software developers seem to have higher expectations when it comes to collaboration and co-development on data analytics and coding. Being able to access and edit content with multiple users at the same time (comparable to e.g. Google Docs) seems to become the standard. Also, freely available data science tools like Google Collaboratory and Kaggle notebooks are raising the bar for what users expect. Regarding the usefulness and ease of use of the offered virtual research components, we saw that in general, participants tend to prefer staying with components that they are already familiar with. The absence of such familiar components in any research environment will increase the learning curve and the required time investment, and thus might be a potential barrier for adoption. In our VRE evaluation this issue was most clearly signalled for data visualisation functionality, where options in many broadly used analytics environments are much more functional and flexible than dedicated visualisation components that are currently offered by AGINFRA PLUS.

The opinions on openness and FAIRness of the VRE varied among the evaluation participants (Figure 5.7). In general, participants agreed that the offered VRE assets for publishing data, algorithms and models are an effective way to foster more open research

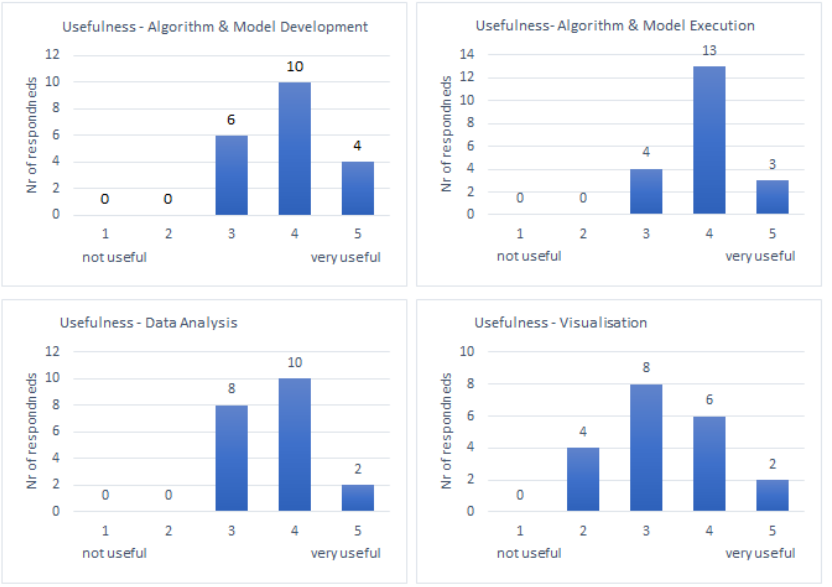


Figure 5.6: Evaluation survey responses for the topic "usefulness".

ecosystems. When it comes to FAIR, participants see the potential of the VREs' publishing functions and catalogue to improve the discoverability and accessibility of data and algorithms, which would also increase the likelihood of other researchers reusing them. At the same time, the still limited support for interoperability was noted as a potential weakness, where several participants particularly criticised the lack of functionality to effectively use semantics and linked data.

Determining a good perception of the learning curve of a research environment proved to be difficult for many of the participants. Most evaluators estimated a relatively steep learning curve toward being able to effectively use the provided VRE. For instance, they quite consistently estimated the efforts required to configure and run an algorithm or model and to analyse its outputs as relatively high, while also acknowledging that this is to be expected from any environment that integrates such a broad range of data science tools.

5.4.1 VRE validation results

The VRE validation, using the final version of the developed AGINFRA PLUS VRE, was performed with a larger and broader group of participants. While the evaluation targeted experienced researchers and data scientists, the validation included potential VRE users from more diverse backgrounds and levels. The setup of the validation was a

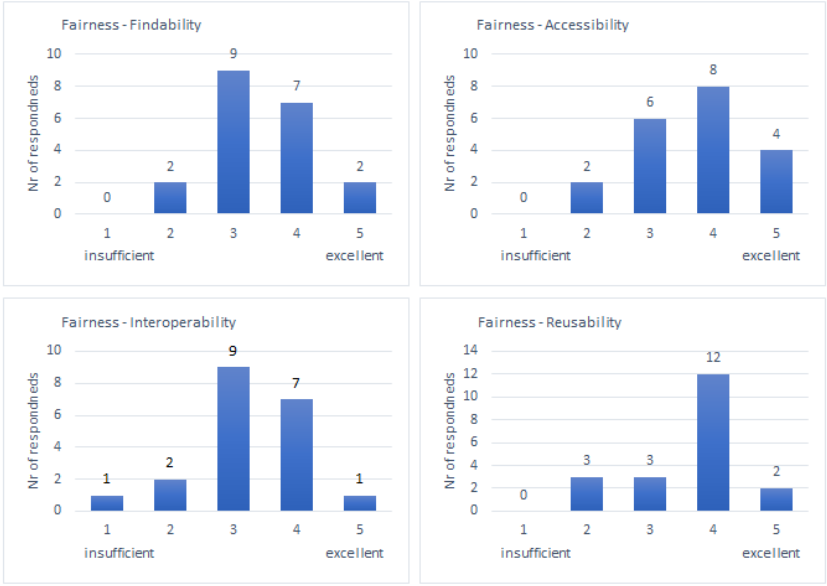


Figure 5.7: Evaluation survey responses for the topic "FAIRness".

one-hour webinar, where participants received a presentation of the VRE and its main components and some practical examples of how the VRE was applied in research case studies were demonstrated. In contrast to the evaluation, the validation did not include the opportunity to perform hands-on work with the environment. The objective of the survey was in this case much more targeted at assessing how participants perceive working with such an environment, how they expect they could benefit from it themselves and how a VRE fits and could be adopted in their organisations. The webinar was attended by 133 participants, of whom 42 have responded to the validation survey.

The results of the validation survey (included as Appendix A) show that respondents see good opportunities in the VRE supporting them to get more productive, and from a personal perspective are motivated to work with such environments. This positive attitude is underpinned among others by the score on a 1–5 Likert scale (Allen and Seaman, 2007) of responses on the following survey questions: Using such a virtual research environment would increase my productivity (mean score = 4.21); A virtual research environment makes work more interesting (mean score = 4.20); It would be easy for me to become skilful at using such a virtual research environment (mean score = 4.07).

The validation survey also revealed some potential barriers to the introduction of VREs as part to the research process. It becomes clear from responses that lack of sufficient knowledge, lack of support for capacity building and lack of resources could impede the

adoption and uptake of such environments. Moreover, there was quite a spread in the perception regarding whether organisations and influential persons within organisations would be motivated to adopt and would support the introduction and use of VREs.

In general, respondents answered positive regarding their motivation and the opportunities they see to start using a VRE. This is among others supported by a mean score of 3.83 on the statement "I intend to use such a virtual research environment in the next 12 months". Remarkably, there were relatively more respondents in the middle range of the Likert scale compared to the statements on personal motivation and opportunities. This might be partly related to the potential barriers that some see to get such environments adopted in their organisations.

5.4.2 Developer experiences

In general developers have, over the three years of the project, experienced a gradual improvement of the support for integrated agro-climatic research workflows by the AGINFRA PLUS VRE. This was mainly due to the adopted iterative approach, and the opportunities it provided to adapt the VRE to feedback and new insights from developers and users. Many requests for change were implemented on feedback of the developers, improving particularly the functions to develop, deploy and run algorithms and to publish, share and reuse research assets. There is a general agreement that the AGINFRA PLUS VRE provides a safe and protected collaborative working environment, while at the same time allowing to access and share resources, like data, algorithms and scientific knowledge and contributing to the objectives of the European Open Science Cloud. As part of the co-development process, the "social media components" in the environment offer effective ways to communicate and stay connected. Regarding data analytics and modelling, the AGINFRA PLUS VRE, supported by the underlying D4Science platform, has proven to be useful to deploy reusable models and to run these on a computing cluster, although in some cases adapting software to the specifics of the platform required substantial efforts.

While the VRE provides several semantic tools, these are up till now insufficiently integrated. It appeared not possible to semantically tag the delivered research assets making use of available vocabularies and ontologies, which is generally considered to be a minimum requirement for FAIR sharing. The VRE's data visualisation components proved not to be fit-for-purpose for the agro-climatic research use cases, lacking much of the required functionality, configurability, and integration hooks. Like the semantic components, individual tools were available, but the lack of integration did not allow to link them into practical research workflows. In practice, VRE developers in the use cases have either developed visualisation components from scratch (e.g. the visualisation dashboard) or used external libraries (e.g. in Jupyter Notebooks). Conclusively, a general observation is that while the composition of a VRE using the D4Science platform consisting of individual components is technically relatively simple, the integration of these components in such

a way that they are useful to setup effective research processes is still a barrier to fully exploit these tools as part of the AGINFRA PLUS VRE, at least for the agro-climatic use cases examined.

5.5 Discussion

Researchers in the agro-climatic research domain have experienced the shared resources offered through the D4Science powered AGINFRA PLUS VRE, from the options to collaboratively experiment in a safe environment and eventually from the added value of deploying, publishing, and reusing developed algorithms and data sets. Virtual Research Environments offer the potential to facilitate collaboration and support open science, also in the domain of agro-climatic research, by digitising and integrating many of the research tools and processes that researchers regularly use in their work. Through the described experiments we have tested this hypothesis for two characteristic use cases in the agro-climatic domain, and we have evaluated the perception and experiences of researchers. On the positive side, first, our results demonstrate that it is feasible to implement workflows that automate and virtualise scientific processes in the agro-climatic domain in online environments. Second, researchers testing the workflows also understood the potential that these VREs have for accelerating their work, while opening new avenues for research, and generally voiced high expectations of such environments in our evaluations. Third, VREs offer the possibility to connect data and algorithms more easily, once the use case has been clearly defined, thereby enabling a step towards FAIR data sharing, reusable algorithms and models, and open science in general. Thus, the potential for added value of VREs in agro-climatic science is clearly identified.

However, there are also a few important trade-offs to consider in the further development of virtual collaboration tools between agro-climatic researchers to facilitate the transition towards more digital agriculture: (1) investing in analysis ready platforms or investing in generic tools that can be easily combined; (2) interoperability of data and tools; (3) the need for an innovative VRE or a stable VRE based on proven technologies. We will elaborate these trade-offs in the following paragraphs.

First, there is a dilemma in either developing platforms that combine tools as a ready-made solution or deploying generic tools and leaving the combination of tools into workflows to the researchers. Evaluation results showed that participants were sometimes doubtful about the use of specific VRE components and preferred different solutions that fitted better to their personal skills and experience. It suggests that the most appreciated functionality is the easy availability of generic tools for collaboration and sharing (i.e. GitHub, Python/R notebooks, data and analytics repository connections) which are cross-platform and interoperable. Researchers in the agro-climatic domain tend to have their own preferences for tools and platforms but want to share the data and analytics through such generic tools. An important implication is that researchers will not accept

platform specific lock-ins on their analytics, making these difficult to share with other researchers. After numerous pleas to share code of analytical solutions (Goldacre et al., 2019; Nature Editorial Board, 2018), agro-climatic researchers are keen to embrace this practice, however, vendor-lock-ins in platforms then do not work. A challenge for platforms is that the integration of all the tools on the platform and the data sources need to be continuously updated and improved as new versions of tools become available or data definitions are upgraded. Platforms are thus always catching up on such developments, with the risk of losing users if the changes are not made quickly enough. Thus, when VREs are conceived as comprehensive platforms that fully automate the researcher's processes in the agro-climatic domain, they are unlikely to succeed as the provided solutions are very vulnerable to maintenance issues and risk locking in researchers into solutions that they do not want. There is, however, potential for VREs to deliver generic tools for sharing of analytical code and components and data connections that facilitate the work, without striving to fully automate it.

Second, improvements in interoperability are required. Evaluation results clearly showed that researchers are seeking ways to improve interoperability toward better reusability of data, algorithms, and models. The need for tools to support semantic linking of (meta)data was mentioned often, as well as the tested VREs being relatively weak in their support for interoperability. Uses cases in this domain are commonly interdisciplinary, combining aspects of domains with a different character and making semantic linkage and interoperability a complex and laborious challenge. In agro-climatic research the user communities are relatively small, and they often employ a systems perspective, combining the different data sets and scientific sub-fields together in one approach. In a way VREs as platforms might offer an opportunity to facilitate researchers with this systems perspective as they bring together tools and data from different scientific subfields. At the same time, involved user communities cannot easily make progress in establishing the conditions required and, for example, lack suitable ontologies to develop interoperable resources. This hampers progress on interoperability, even if the supporting tools are available.

In the cases presented here, the platform solutions are developed for a niche group of researchers, as agro-climatic research falls into many diverse specialised segments studying different aspects of agro-climatic systems. To illustrate this further, the agro-climatic research community is at the intersection of three research communities, i.e. climate science, agriculture, and Earth Observation. This is characterised in general terms with respect to aspects of Big Data, (Table 5.2), even if that is not representative of all research happening in these communities.

In climate science, the data sets describe physical and natural phenomena, and have been measured, or are modelled for future predictions or long-term scenarios. The complexity of the data is manageable (even though data sets can be very large) and quite easily linkable (since they represent globally agreed upon physical phenomena). Besides, data

Table 5.2: Characterising climate, agriculture, and Earth Observation sciences along four Vs of Big Data.

Research community	Volume	Velocity	Variety	Veracity
Climate	*High*: Large size gridded data sets (e.g. in NetCDF) with high temporal resolution.	*High*: New forecasts and observations are produced very frequently.	*Low*: Data types are consistent over data sources.	*Medium*: A lot of focus on forecasting and future scenarios.
Agriculture	*Low*: Usually smaller data sets (excluding weather data) that can be managed on a single computer	*Low*: Data collection is usually in pace with crop growing cycles, except for some developments around IoT devices.	*High*: many diverse data types that can be collected, following a wide range of agricultural systems across sectors and regions.	*High*: Uncertainty on underlying agricultural system functioning; application of expert rules.
Earth Observation	*High*: Imagery with e.g. multiple spectral or wave bands covering large regions in high resolution.	*Medium*: Every day or few days new imagery becomes available for a region.	*Medium*: A limited number of spectral or wave bands, however, converting them to e.g. indices for applications introduces diversity.	*Low*: Uncertainties come from different algorithmic approaches in the conversion of the indices.

are usually already shared between researchers due to the size and costs for collecting it, and a relatively well-developed open culture. Climate researchers are used to work with computational clusters such as available for High Performance Computing (HPC) or setting up a specific calculation workflow using NetCDF files based on hardware restrictions of individual servers or computers. This community offers good potential for VREs, as there is a need for computational resources, and workflows can be standardised.

In contrast, in agricultural research, there is a lot of diversity of data as agricultural systems are wildly diverse across sectors and geographies. Often only a specific aspect of the agricultural system is studied, making modelling choices and assumptions, and thus with a difficulty to link data sets to one another. A lot of research is built on discovering relationships on how agricultural systems function as a living environment, and often there are a lot of soft-knowledge and expert rules applied to establish relationships when the underlying mechanisms are too complex. As an implication, data sets are also heterogeneous, making it difficult to link them together on seemingly similar data types. This might also explain the concern expressed by many evaluators about the lack of operational functions available in the VRE to use semantics to link data sets. In addition, access to data, both within the sciences and in the private or public sector, is a further obstacle (Wolfert et al., 2017). There is generally less of a sharing culture, also because research groups are relatively small and connected. Moreover, researchers often work with the data on their own computers and might share with colleagues based on their preferences (potentially competitive). There are few automated workflows, as every research question

requires a different approach. This is exemplified in the use case with the WOFOST crop growth model implemented in the AGINFRA PLUS VRE, which is set up to study a specific case. If a different case is studied, for example related to soil fertility or nutrient management, the model and the data set will need to be changed and there might be expert knowledge required to link crop growth to nutrient management.

Finally, Earth Observation (EO) research focusses on the use of images taken by satellites, drones, and other airborne sensor platforms to monitor and understand natural phenomena, for example by deriving indices based on the different bands in a multispectral image. Due to a basis in open access satellite data through e.g. the NASA Landsat missions (https://www.usgs.gov/landsat-missions) and the Copernicus Earth observation programme of the European Union (https://www.copernicus.eu/en) there is a widespread common practice of making research results in terms of data available to others. In recent years, through programmes such as EU's Copernicus, more data became available with finer spatial resolution and higher acquisition frequency. Each sensor requires its own calibration and data processing algorithms, however the starting point, spectral or wave bands, is often the same. This is shown by the second use case in the AGINFRA PLUS VRE in which Sentinel-2 data is used to estimate the crop phenology and growth curves of different agricultural crops. If another sensor was to be added, for example, a drone-based camera, it needs to be calibrated against the already available sensor.

This is just one example of often many expert steps required before such data can easily be used in a plug-and-play fashion. In case of more automated, generic procedures and workflows, implemented as part of a VRE, there will be more data loss as only those data that fall perfectly in line with data from other sensors can reliably be used. Also, VREs do not necessarily provide access to the expertise needed to use it properly. Hence, these interoperability challenges are an obstacle to the further adoption of collaborative environments in agro-climatic research, which is partly due to the fragmented nature of the research field and to the weakness of VREs in general to support this. These interoperability challenges are not limited to scientific collaboration tools, but also applicable to FAIR data (Top et al., 2022), and e.g. Farm Management Information System (Fountas et al., 2015; Tummers et al., 2019), so they pose a larger challenge to the implementation of digital agriculture and systems thinking in agro-climatic science.

Even though our use cases encompassed all three scientific fields, they were focused on an analysis of the agricultural system, using data from climate science (e.g. weather records) and earth observation (e.g. spectral indices derived from Sentinel-2 data). In hindsight, use cases that focused more on the climate or Earth Observation might have been more promising as there are (1) more common practices of working with remote computational resources; (2) more familiarity with sharing access to research data; (3) basic sets of data types that are more easily linkable across data sets compared to agriculture science. In a way our findings reinforce the argument made by Allan, 2009: "Many VRE implementations

revolve around finding niche solutions to sometimes very niche problems faced by a small group of researchers working in a particular field of study".

A third trade-off exists in the role that VREs play in the wider trend towards open science. Ultimately, VREs represent a possible end-state of open science once it goes beyond opening the scientific process in terms of data and algorithms. VREs are then becoming an integrated collaboration and productivity tool, not only making the data and algorithmic resources available, but also connecting researchers in common workflows and conceiving innovative working environments (Thanos, 2013). Given their still innovative nature and potentially more conceptual set up, VREs might not yet have a large day-to-day role to play in open science practice. Open science is more about operationalising good and proven practices of sharing than about experimenting with (immature) innovations. This might also confuse researchers on the expectations of them in working according to open science practices. What VREs and open science have in common, as argued by Allan, 2009 and also found in our evaluations and validations, is that they need to take into account the 4Cs to become a common practice: *Culture of research embracing new methodologies; Champions in open science; Communications to disseminate the added value; and integrated Change management to make it happen.*

5.6 Conclusions and recommendations

Based on the evaluation results, and analysis of the use cases, conclusions have been reached on the four research objectives, as documented in Table 5.3.

Ultimately, agro-climatic research will gradually deploy more digital techniques to further the science itself, increasing reproducibility and transparency. This represents a gradual transition towards digital research, and to facilitate this transition, the recommendations have been formulated against the different stakeholder roles, as listed in Table 5.4.

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under AGINFRA PLUS project (grant agreement No. 731001).

❄

Table 5.3: Overview of the research objectives and conclusions.

Research objective	Evaluation results and insights
What are the requirements of agro-climatic researchers in their research tasks, specifically to support Open Science?	Researchers are open to share their data and algorithms if this is made easy. They want to avoid adapting too many of their working practices to work with a new environment.
What would be the required functionality for a VRE to be successful in fulfilling these requirements?	A VRE needs to offer generic tools and functionalities that benefit the agro-climatic researchers to do their work faster, and help them to publish data and tools, while linking to existing data.
How can agro-climatic researchers benefit from a VRE and what are the implications for their work?	Agro-climatic researchers experience benefits if the data processing steps they want to perform are fully established, so that they can be incorporated in VRE workflows, or for sharing algorithms, models and data sets.
What would be required from researchers and research organisations to adopt a VRE?	VREs would need to be well integrated into the day-to-day working practices of the researchers and research organisations, and have a large enough base in potential users to be usable and become self-sustainable with respect to development and funding.

Table 5.4: Final recommendations.

Role	Recommended activities
Research managers	Ensure that researchers can work on the science itself, not on the development of technologies to do the science, thus offering proven technologies fitting into the daily working practices of researchers.
Researchers	Embrace good practices in opening of research products, including data, models, and tools, where the agro-climatic domain in some respects is lagging.
VRE developers	Focus on making those tools available that are proven solutions. Technologically driven innovations are less relevant to achieve scientific progress.
Research funders	Support the transition in their funding models, most notably ensuring that researchers make research products openly available, and by focusing less energy on the technological innovation while applying (as with VREs and FAIR) and building resilient and adaptive scientific communities which can easily embrace new developments.

Appendices

5.A Results of the validation survey

Table 5.A.1: The five-point Likert scale used in the survey for all questions

Strongly disagree	1	2	3	4	5	Strongly agree

Table 5.A.2: Survey results of the validation webinar

Question	1	2	3	4	5	N/A	N	Mean Score
1. I Would find such a virtual research environment useful in my job.	1	0	2	12	27	0	42	4.52
2. Using such a virtual research environment would enable me to accomplish tasks more quickly.	1	0	2	21	18	0	42	4.31
3. Using such a virtual research environment would increase my productivity.	1	0	4	21	16	0	42	4.21
4. If I used a virtual research environment, I would increase my chances of getting a better position or salary.	0	4	18	12	8	0	42	3.57
5. My interaction with such a virtual research environment would be clear and understandable.	1	2	11	20	8	0	42	3.76
6. It would be easy for me to become skilful at using such a virtual research environment.	1	1	6	20	14	0	42	4.07
7. I would find such a virtual research environment easy to use.	0	2	17	15	8	0	42	3.69
8. Learning to operate such a virtual research environment would be easy for me.	1	2	12	19	7	1	41	3.71
9. Using such a virtual research environment is a good idea.	1	0	0	14	26	1	41	4.56
10. A virtual research environment makes work more interesting.	1	1	5	16	18	1	41	4.20
11. Working with such a virtual research environment is fun.	0	1	10	15	14	2	40	4.05
12. I would like working with such a virtual research environment.	0	1	3	8	5	25	17	4.00
13. People who influence my behaviour would think that I should use such a virtual research environment.	1	7	12	15	7	0	42	3.48
14. People who are important to me would think that I should use such a virtual research environment.	1	8	11	14	8	0	42	3.48
15. The senior management of my organisation would be supportive of using such a virtual research environment.	0	2	17	15	7	1	41	3.75
16. In general, my organisation would support the use of such a virtual research environment.	1	2	12	19	8	0	42	3.74
17. I have the resources necessary to adopt and use such a virtual research environment.	4	11	9	12	4	2	40	3.03
18. I have the knowledge necessary to adopt and use such a virtual research environment.	3	7	11	12	8	1	41	3.37
19. The virtual research environment does not seem compatible with other systems I use.	4	9	18	6	4	1	41	2.93
20. In my organisation, a specific person (or group) would be available to assist me with difficulties in using such a virtual research environment.	8	9	18	6	4	1	41	2.76
21. I could complete a job or task using this virtual research environment if there was no one around to tell me what to do as I go.	3	7	18	8	5	1	41	3.12
22. I could complete a job or task using this virtual research environment if I could call someone for help if I got stuck.	2	2	13	16	8	1	41	3.63
23. I could complete a job or task using this system if I had a lot of time to complete the job for which the software was provided.	3	3	13	13	8	2	40	3.50
24. I could complete a job or task using this virtual research environment if I had in my organisation a facility for assistance.	2	1	12	16	9	2	40	3.73
25. I feel apprehensive about using such a virtual research environment.	11	5	9	11	3	3	39	2.74
26. It scares me to think that I could lose a lot of information using such a virtual research environment by hitting the wrong key.	13	15	9	2	2	1	41	2.15
27. I hesitate to use such a virtual research environment, fearing to make mistakes I cannot correct.	12	17	8	2	2	1	41	2.15
28. Such a virtual research environment looks somewhat intimidating to me.	16	14	7	2	2	1	41	2.02
29. I intend to use such a virtual research environment in the next 12 months.	1	2	14	10	14	1	41	3.83
30. I predict I would use such a virtual research environment in the next 12 months.	1	2	13	14	11	1	41	3.78
31. I plan to use such a virtual research environment in the next 12 months.	1	2	14	11	13	1	41	3.80

Chapter 6

Synthesis

6.1 Main findings

This thesis focusses on data engineering for digital agriculture, specifically related to the work with geospatial data. In general, data engineering involves the design, construction and maintenance of systems that enable the collection, storage, and analysis of large volumes of data. For geospatial data and modern digital agriculture the 'V's (Volume, Velocity, Variety, Veracity) that characterise Big Data quickly can become a concern and need to be taken into account. When done effectively, it can further drive digital transformation in agriculture by enabling and improving data-driven decision-making processes.

Although spatial data engineering and spatial data science continue to evolve with rapid technological advancements, the core principles established in this research remain fundamental for future innovations in digital agriculture. Specifically, it has been shown that: (i) connecting data; (ii) connecting models; (iii) connecting systems; and (iv) connecting researchers are beneficial to the required data engineering and thus to the data science as well, following the classic *data–information–knowledge–wisdom* route. Table 6.1 provides an overview.

Table 6.1: Geospatial data engineering for digital agriculture & the DIKW levels

Topic	DIKW Level	Data engineering aspects	Example in digital agriculture
Connecting data	Data	Geospatial data ingestion and storage	Integrating satellite imagery (Sentinel, Landsat) with IoT farm sensor data.
Connecting models	Information	Geospatial data processing and modelling	Combining weather, soil, and crop models for better analytics.
Connecting systems	Knowledge	Pipeline orchestration, scalability, and performance	Running large-scale models on (cloud-based) infrastructure.
Connecting researchers	Wisdom	Governance, collaboration, and decision support	Sharing actionable insights across digital agriculture stakeholders.

Considering the research objectives of this thesis, the key findings are as follows.

Chapter 2 concludes that Variety and Veracity are the most interesting to focus on from the perspective of spatial data engineering for agriculture. Veracity ("trust in the data") plays a part in Chapter 5, by using Virtual Research Environments (VREs) to provide online community workspaces with trusted data sharing. Other solutions (e.g. blockchain, data spaces, federated learning) are interesting and suggested for future research.

The variety of data can be split into two major parts: semantics and spatio-temporal characteristics. Solutions to address data semantics are known for a long time (e.g., ontologies, resource description framework, SPARQL, and triple stores) but not so often applied: (i) ontology engineering requires specific skills and is not an easy task; (ii) working with knowledge concepts and their relations, broken down into elemental Resource Description Framework (RDF) *subject–predicate–object* statements, is not part of mainstream technology and software engineering; (iii) semantic tools and technologies are sparse, with

open source solutions of lesser quality compared to the few expensive commercial software available; (iv) there is a lack of usable globally linked agriculture ontologies, and given the domain, these would quickly extended into other domains and sciences, making them even harder to establish.

Related to spatial aspects of data, there is extensive standardisation rooted in Geographic Information Systems (GIS) and supported by national Spatial Data Infrastructures (SDIs). These frameworks offer well-established mechanisms for representing geographic features, including vector and raster data formats, coordinate reference systems, and spatial metadata. However, these spatial standards do not necessarily overlap with semantic standards. A key distinction lies in how spatial information is modelled. In GIS standards such as INSPIRE (2024) or GML (2018), spatial entities are defined primarily through their geometry–*a polygon is the field*. In contrast, semantic models typically describe spatial characteristics as properties of conceptual entities–*a field hasGeometry polygon*. This geometry-first approach contrasts sharply with the entity–property–relation structure of semantic models and is one of the reasons why spatial and semantic interoperability remains a complex engineering challenge.

For example, mapping a Global Positioning System (GPS) coordinate from an agricultural data set to field geometries or vegetation indices (e.g., NDVI) stored in raster grids requires more than just coordinate transformation and resolution alignment. It involves reconciling different assumptions about how entities and their spatial attributes are structured, labelled, and linked between standards. Bridging these paradigms demands deliberate engineering to ensure consistent and meaningful integration of spatial information.

Although this thesis focusses primarily on spatial aspects, temporal interoperability presents a parallel challenge. For example, in agricultural systems, the use of decadal time units is widespread, whereas GIS platforms typically rely on timestamp-based models. This discrepancy in temporal granularity also introduces a semantic gap that further complicates data integration. Recent studies highlight the importance of harmonising spatial and temporal dimensions to enable reliable data fusion and time-sensitive decision support in digital agriculture (San Emeterio de la Parte et al., 2024; San Emeterio de la Parte et al., 2023; Zeginis et al., 2024).

Both aspects of variety are relevant when connecting models and data sources at the information level, as discussed in Chapter 3. It examined the usefulness of a standard (OpenMI) for model linking that supports both semantics and spatio-temporal characteristics of data in a number of research domains. Although such a standardisation is appreciated (by researchers), its use should not be enforced, and the specification should define a comprehensive, self-contained set of functionalities and protocols, ensuring that compliant implementations possess all essential components, since research projects usually do not contribute to the development of frameworks and common tools.

In Chapter 2 the development of technologies that can address the Volume and Velocity of Big Data is expected to be taken up by industry. Chapter 4 examines the usefulness of one of such available solutions, the open-source Apache Spark analytics engine, applying it to the running of large-scale crop growth simulations with a mechanistic model (WOFOST). It is successful, but cannot avoid that scalable/distributed systems are inherently more complex than non-distributed implementations. Scalable systems involve multiple computers working together over networks and thus have to be able to handle failures and stay in sync even when parts break or network communication gets delayed. This also affects software engineering, especially when optimising for performance and low environmental resource use in such settings.

Due to this increased complexity, agronomists might not adopt big data technologies and instead accept long run-times or choose to only work with scaled down data. However, since these tools are standardised, data engineers can gain experience with them and apply their knowledge across projects. The functionality can then be made available to agronomists and other researchers and stakeholders as part of a Virtual Research Environment, as has been studied in Chapter 5. This research shows that VREs are an effective solution for enabling scientists to collaboratively access, manage, and analyse large, complex data sets, using computational resources from different locations, in an integrated digital workspace. VREs can, however, be difficult to maintain, e.g. after a research project has ended and there is only a small user community. Similarly to model integration frameworks (discussed in Chapter 3), VREs have to provide all tools needed by the researchers, or at least integrate well with the tools they are already familiar with.

6.2 Reflections

6.2.1 On the use of semantic technologies in digital agriculture

During the period covered by this thesis and its referenced literature, interest in using full formal ontologies as part of semantic web technologies–such as the Resource Description Framework (RDF) and the Web Ontology Language (OWL)–first increased and then declined in the context of digital agriculture. With an ontology defined as an “explicit specification of a conceptualisation” (Gruber, 1995), it becomes a formal ontology when this specification is expressed in a machine-interpretable, logic-based form. A full formal ontology is a highly structured and logically rigorous representation of knowledge within a domain, defined using formal languages and logic-based semantics, typically aiming for consistency (no contradictions), completeness, and automated logical reasoning. This is in contrast to the recently preferred lightweight or ”informal” ontologies.

The use of ontologies and other semantic technologies in digital agriculture, starting with precision agriculture, has long been considered a critical enabler of interoperability, that is, the solution to the challenge of data heterogeneity, especially in the domain of

spatial data engineering. Semantics offers structured vocabularies–such as the ICASA data standards (White et al., 2013)–and ontological scaffolds that can bridge gaps between data variety (e.g. Rijgersberg et al. (2011)), linked models, and decision-making systems. From the early 2000s domain ontologies such as AGROVOC, Crop Ontology, and the Semantic Web for Earth and Environmental Terminology (SWEET), promise to unify agronomic, environmental, and geospatial data sets. A promise that continues to prove to be difficult to scale in practice. Ontology development is resource intensive, triple stores and reasoning tools are not sufficiently performant for real-time applications, and uptake outside academic projects is difficult and limited.

The initial wave of semantic integration in environmental and agricultural sciences–prominent from the early 2000s through the mid-2010s–was marked by ambitious efforts to develop formal ontologies for comprehensive knowledge representation. The projects aimed to establish rigorous logic-based systems that could unify data across disciplines, institutions, and geographic regions. These initiatives were fundamental in advancing the vision of the semantic web for integrative modelling and data sharing (Athanasiadis, 2015; Athanasiadis et al., 2009; Rizzoli et al., 2008; Samourkasidis and Athanasiadis, 2020; Villa et al., 2009).

Despite their conceptual strength, full ontology-based approaches encounter significant challenges in real-world application. Ontology engineering required intensive collaboration between domain experts and knowledge engineers, and the maintenance of formal structures proved difficult in the face of rapidly evolving scientific domains. Furthermore, reasoning engines and semantic query languages were often too slow, complex, or poorly supported to integrate with operational workflows in agriculture and environmental research. Specifically for spatial data, it took a while before *GeoSPARQL* became the unified standard through the Open Geospatial Consortium (OGC). Its focus is vector data; the combination of semantic technology and spatial raster data remains an ongoing active area of research (Hamdani et al., 2023). Although formal ontologies were originally intended to enhance clarity, transparency, and integration in data-driven systems, such as decision support systems, their implementation has often resulted in unintended opacity for the users. In practice, full, formal ontologies–particularly those based on description logics (Horrocks, 2005)–can become *epistemic black boxes* that obscure rather than illuminate reasoning.

In the mid-2010s, a clear shift was underway toward more lightweight, pragmatic technologies that emphasised flexibility, usability, and *partial semantics*. This transition was driven by the need to enable rapid data exchange, integration, and reuse without the overhead of formal ontological commitments. In this new paradigm, technologies such as JSON for Linking Data (JSON-LD), the schema.org open community driven vocabularies, and simple *RESTfull APIs* became preferred tools for implementing semantic interoperability. The goal with these lightweight approaches is not full logical inference, but rather mutual understanding and data discoverability. Instead of relying on formal axioms and deep

class hierarchies, these systems encode just enough structure—through controlled vocabularies, metadata standards, and simple linked data patterns—to support integration and interpretation across platforms.

The rise of *FAIR data movement* (Wilkinson et al., 2016) further accelerated the adoption of lightweight semantics. The FAIR principles–Findable, Accessible, Interoperable, and Reusable–encourage data providers to use standard identifiers, ontologies, and metadata schemas in ways that are machine-actionable but not overly rigid. This balance has led to widespread use of shared vocabularies (e.g., DCAT (W3C), AGROVOC (FAO), Croissant (ML Commons)), ontology portals (e.g., AgroPortal by the Agrisemantics Working Group of the Research Data Alliance), and metadata catalogues (e.g., CKAN, GeoNetwork, STAC – SpatioTemporal Asset Catalogs) in both research and development contexts.

In practice, lightweight semantic technologies have enabled greater agility and participation in data-sharing initiatives. This shift represents not a rejection, but an evolution of the semantic vision. The full formal ontologies laid the theoretical groundwork for machine-understandable agriculture and environmental science. Lightweight technologies have extended that concept into practical, scalable solutions that are more closely aligned with the pragmatic realities of spatial data engineering, multi-stakeholder collaboration, and rapidly evolving digital ecosystems. Unfortunately, the shift also enforces existing limitations. Following the flow and terminology of the DIKW pyramid, much of digital agriculture risks remaining trapped in the lower layers–data and information–with limited progress toward machine-actionable knowledge and contextualised wisdom (actionable insights), the layers where the *interoperable* aspect of FAIR is increasingly relevant.

Semantic technologies in some form, especially when coupled with spatial metadata standards, both preferably modern, lightweight, and practical, remain critical for lifting digital agriculture beyond siloed data aggregation into integrated, knowledge-rich systems.

6.2.2 On mechanistic models and AI in digital agriculture

Artificial Intelligence (AI) and Machine Learning (ML) techniques have become indispensable in digital agriculture (Kamilaris and Prenafeta-Boldú, 2018; Liakos et al., 2018; Sharma et al., 2021), as also referenced in Section 4.4. This introduces a new and arguably more critical appearance of *epistemic black boxes* in the domain. Although these technologies provide powerful predictive capabilities and automation, they often lack transparency, explainability, and causal grounding–qualities essential for scientific legitimacy and responsible decision making.

Most modern machine learning models, particularly deep learning, operate as high-dimensional, non-linear statistical approximators of the underlying function or relationship mapping inputs to outputs in complex data. They excel at identifying patterns in large data sets, but typically do so without revealing the underlying rationale behind their

predictions. For example, fertiliser adjustments might be recommended to a farmer, without any agronomically interpretable explanation. This opacity again poses a serious risk to scientific understanding and decision making. Without sufficient insight into the internal workings of a model, researchers and practitioners cannot assess whether outputs are valid, biased, or generalisable. Moreover, current AI models frequently *optimise for correlation rather than causation*, capturing superficial relationships in historical data that may not hold under changing climatic, social, or ecological conditions. In the real world machine learning models are faced with e.g. data drift and concept drift, which needs to be monitored and the models retrained to maintain prediction accuracy.

Another concern is the hidden nature of data biases. Since AI/ML systems learn directly from training data, they may perpetuate or amplify historical inequities, regional imbalances, or data collection artefacts. These biases are often hard to detect due to the opaque architecture of the models, creating a double layer of black-box behaviour–both in logic and data provenance. The *Garbage In, Garbage Out* effect already applies to mechanistic models, but it is an even bigger concern for AI and ML models.

Table 6.2 shows the parallels between the black box issues when introducing full formal ontologies or non-explainable AI in digital agriculture (see also Section 6.2.1).

Table 6.2: Black Box aspects of formal ontologies and non-explainable AI in digital agriculture.

Aspect	**Formal Ontologies**	**Non-Explainable AI**
Black Box Cause	Logical inference chain	Statistical/mathematical abstraction
Risk	Misunderstood formalism	Unjustified correlations
Barrier	Requires ontology expertise	Requires ML/AI expertise
Transparency	Low without logic tracing	Low without explainability techniques
Validation	Challenging for non-experts	Difficult even for developers

There is a growing movement to address these concerns through *interpretable and eXplainable AI (XAI)* techniques–such as Shapely values, Local Interpretable Model-agnostic Explanations (LIME), and causal inference–but these are not yet widely adopted in digital agriculture, environmental sciences, and Earth Observation in general (Tuia et al., 2024).

In terms of the DIKW pyramid, AI–sub-symbolic in particular–can be seen as an attempt to *short-circuit* the classical flow by bypassing some of the intermediate steps (especially the human-driven interpretation between levels). Instead of relying on structured *data → information → knowledge → wisdom* transitions, AI often tries to jump directly from raw data to decision making, e.g., through deep learning and reinforcement learning. This approach has both advantages and disadvantages:

i. Loss of explainability and trust: The classical DIKW flow builds knowledge incrementally, allowing experts to verify information at each stage. Artificial intelligence,

particularly deep learning, operates as a black box, making it difficult to justify decisions in agriculture.

ii. Lack of incorporation of domain knowledge: Classical DIKW integrates, e.g. mechanistic models that ensure consistency with agricultural science. AI can sometimes detect spurious correlations in the data and base predictions on those.

iii. Over-reliance on data: AI models, if trained on biased or low-quality noisy data, can produce misleading results. Classical DIKW transforms raw data into validated information before it is used in decision making.

iv. Ethical and social risks: AI-driven decision-making can lead to unintended consequences, when human oversight is lacking. In classical DIKW the human oversight is built into the flow.

A hybrid classical DIKW flow including AI will therefore be more effective for digital agriculture. For example, using ML for pattern recognition but validating or explaining it with domain knowledge (i.e., by agronomists, or a self-explainable symbolic – sub-symbolic AI model Höhl et al., 2024). Or by combining mechanistic models with AI (hybrid modelling, or physics-informed modelling) for better accuracy and real-world consistency. In general, using human-in-the-loop AI approaches and systems to increase trust and explainability and avoiding full AI automation. Furthermore, to avoid technological push, specifically in ML/AI, a human-centric design and development approach is to be preferred (see next section).

Ultimately, science and decision making demand not just predictive power, but also understandability, accountability, and reproducibility. As digital agriculture moves toward increasingly AI-driven systems, ensuring that these principles are preserved is both a technical and ethical imperative for spatial data engineering.

6.2.3 On the need for human-centric design in digital agriculture

This research has shown that spatial data engineering plays an essential role in the digital transformations of agriculture. As a discipline, it must bridge geospatial science, (spatial) data science, information science, computer science, and agricultural practice–responsible for structuring, transforming, and integrating diverse heterogeneous spatial data sources to support insight generation and decision-making across farm, advisory, and policy levels. At first glance, spatial data engineering for digital agriculture appears to be a purely technical endeavour that involves complex pipelines, cloud platforms, and algorithms. A central argument of this thesis is that technical excellence alone is not sufficient.

The success of digital agriculture hinges on downstream users—farmers, agronomists, and other stakeholders—trusting and adopting the data-driven systems. In practice, if a data engineering solution is delivered as an *opaque black box*, users may reject it in favour of

local data and tools that they fully understand and control, even at the cost of lower data granularity or output quality.

To mitigate this risk, *transparency and human-centric design* must be built into spatial data engineering–being the foundation for data science–from the ground up, very similar to common software engineering approaches (Knapen et al., 2010; Verweij et al., 2010). This means providing clarity about data sources, processing steps, and the uncertainty or rationale behind recommendations. Approaches such as eXplainable AI are valuable on the data science side, for example, offering explanations for predictions of an AI model so that farmers understand why a recommendation was made (see previous section). Beyond AI models, the entire data pipeline should strive for "white-box" characteristics when possible. For example, instead of simply telling a farmer that "Field 107 needs irrigation," a human-centric system would convey the supporting information: e.g. "Field 107's soil moisture dropped below threshold X based on sensor data at 10 cm depth".

Throughout this thesis, a recurring theme is *connection*–the idea that bridging gaps between data, models, systems, and human experts can greatly improve spatial data engineering and with that the data science outcomes. The research framework followed connects these components to ensure that the flow from raw data to decision is seamless and user-informed. In practice, this could mean interactive systems in which researchers and farmers can inspect model output, adjust parameters, or trace back how the system came to a suggestion. Virtual Research Environments, discussed in Chapter 5, can provide such functionality.

Connecting researchers and domain experts into the loop is perhaps the most critical connection. Instead of a one-way pipeline from data to model to user, a two-way collaboration is needed: domain experts guide data engineering and data science by providing ground-truth knowledge and validating results, and in return the system augments their expertise by providing new insights. This cycle has been shown to improve both the system's performance and the user's trust in the system. For example, incorporating a "farmer-in-the-loop" can leverage the farmer's experience to catch anomalies that algorithms alone might miss, leading to more robust and trusted outcomes. Spatial data engineering and data science, when viewed as a connected ecosystem of technology and people, becomes far more robust and adoption-friendly than a siloed, fully automated black-box approach.

An important insight from this synthesis is that human-centric design and technical sophistication are not opposing forces–they are complementary and indeed codependent for successful digital agriculture. The early visions of Agriculture 4.0 emphasised automation and data on a scale, sometimes at the expense of human involvement. However, experience has shown that purely technology-centric solutions can fail in real-world agricultural settings. It is now increasingly recognised that the next phase – often termed Agriculture 5.0 – must balance technological prowess with human context and needs. In the European Union,

for example, emerging regulatory frameworks for AI in high-impact sectors (including agriculture) explicitly mandate interpretability and human oversight, reflecting the necessity of centring the human user. Far from reducing system efficiency or intelligence, a human-centred approach enhances overall capability. As Holzinger et al., 2024 put it, we should use AI and data systems as "power steering for the brain", which amplifies human decision making rather than trying to replace it.

On multiple occasions, this thesis also highlights the importance of *interdisciplinary collaboration*. Spatial data engineering often serves as a link between traditional GIS practitioners, who prioritise spatial accuracy and standards, and data scientists, who focus on model performance and statistical robustness. Collaboration with domain scientists and agricultural stakeholders brings domain knowledge, contextual insight, and the aforementioned feedback loops that are essential for iterative improvement of systems and processes. A key enabler of this collaboration is the presence of *hybrid roles*–such as agri-informaticians–who can operate across technical, analytical, and domain boundaries. These individuals ensure that spatial data-driven agricultural systems are not only technically sound but also socially and operationally relevant.

This thesis positions spatial data engineering as a human-centric integrative practice that is vital to provide reliable and impactful digital agriculture systems. The transition from data to wisdom is not fully automatic–it must be carefully engineered, validated, and contextualised.

6.3 Societal relevance

Digital agriculture is increasingly recognised as a key driver for transforming food systems toward sustainability, resilience, and inclusiveness (Bertoglio et al., 2021; Food and Agriculture Organization of the United Nations (FAO), 2019). By integrating advanced technologies such as remote sensing, the Internet of Things (IoT), big data analytics, machine learning, and artificial intelligence, digital agriculture allows for more precise, efficient, and adaptive farming practices. These innovations facilitate optimal resource management, early detection of crop stress, dynamic pest and disease control, and customised soil and water management, thus contributing to increased productivity and reduced environmental impacts. In addition, digital agriculture supports broader socioeconomic objectives by enhancing farm profitability, allowing informed decision-making among smallholder farmers, and fostering more equitable access to agricultural innovations. Leading international organisations, including the Food and Agriculture Organisation (2022) and the World Bank (2023), have highlighted the pivotal role of digital agriculture in addressing urgent global challenges such as food security, adaptation to climate change, and sustainable rural development. As such, its societal relevance is now firmly established in the academic,

policy and practice domains, positioning it as a cornerstone of the sustainable development efforts in agriculture.

Spatial data engineering forms the backbone of digital agriculture by providing the means to collect, manage, and analyse geographically referenced data essential for, e.g., precision farming and decision support systems (Obi Reddy et al., 2023). Despite the limited number of studies explicitly assessing its societal impacts, its foundational role within digital agriculture allows for a logical extrapolation of its societal relevance. A comparable situation exists in software engineering, where empirical studies have begun to address human and societal aspects (Garousi et al., 2020; Storey et al., 2020). However, much of the research in software engineering still predominantly emphasises technical innovation, with societal dimensions often considered secondary or analysed only indirectly. This parallel suggests that the difficulty of empirically capturing societal relevance is not unique to spatial data engineering but reflects a broader characteristic of technical disciplines undergoing digital transformation. However, despite these empirical limitations, numerous practical applications already illustrate how spatial data engineering directly addresses critical societal needs within agriculture. Two examples serve to highlight this contribution.

First, spatial data engineering supports the next generation of farmers by mitigating the effects of initially limited hands-on farming experience through advanced, data-driven decision-support systems. The increasing reliance on spatial data-driven tools in agriculture illustrates a direct societal relevance. As Bampasidou et al., 2024 highlight, the emerging generation of farmers exhibits greater technological proficiency than previous cohorts, particularly in using digital and data-intensive platforms. However, this generation often lacks the experiential knowledge and historical agronomic context that traditionally informed agricultural decision-making, a gap that is exacerbated by the uncertainties of a changing climate. Spatial data engineering, by enabling the development of robust, accessible, and context-rich decision-support systems, plays a crucial role in bridging this knowledge gap, thus supporting informed, adaptive, and resilient farming practices for future generations.

Second, it empowers smallholder farmers in low- and middle-income countries (LMICs) by providing localised spatial information and accessible platforms, enabling more informed and resilient agricultural practices. Through the development of user-friendly platforms and the provision of localised spatial data sets, spatial data engineering enables smallholders to access actionable information on critical aspects such as planting, irrigation scheduling, and harvesting. By facilitating informed decision-making at the farm level, these technologies contribute to improving productivity, resource efficiency, and resilience among some of the most vulnerable agricultural communities worldwide (Ceccarelli et al., 2022; Food and Agriculture Organization of the United Nations (FAO), 2019).

While these examples demonstrate the positive societal contributions that spatial data engineering can enable, they also highlight the importance of ensuring that the benefits of digital agriculture are equitably distributed. Without proactive intervention, digital agriculture risks becoming another sector dominated by large service-oriented economies and their private organisations. Table 6.3 outlines some of the possible negative impacts this could have.

Table 6.3: Risks of BigAgri and BigTech for digital agriculture

Risk	Why It's a Problem?	Example
Proprietary AI and digital lock-in	Farmers depend on single-platform ecosystems controlled by corporations.	Closed AI-driven tractors restrict access to repair tools, forcing farmers to buy services from a specific provider.
Data extraction without benefit sharing	Farmers generate valuable agronomic data, but Big Tech owns and profits from it.	AgTech firms collect soil, yield, and weather data but don't compensate farmers.
Unequal access and high costs	AI-based solutions favour large industrial farms over smallholders.	AI-driven precision farming tools require expensive sensors, cloud processing, and high-speed internet.
Market concentration and fewer choices	Small AgTech startups get acquired by Big Agri, reducing competition.	One or two large companies dominate digital seed analytics.

Through concerted efforts by farmers, researchers, and policymakers, it is feasible to establish a modern AI-driven agriculture that is more equitable and sustainable. *Wageningen University & Research must play a crucial role in shaping the future of ethical AI in digital agriculture and in bridging the digital divide* by leading a farmer-centric, human-focused AI transformation, beginning with the development of a robust spatial data engineering foundation for digital agriculture through:

i. Collaborating and actively promoting the development and use of shared (lightweight) semantics in agriculture and related domains.

ii. Contributing to and/or funding open source related work relevant for digital agriculture to increase their survivability in niche markets.

iii. Developing open-source AI and spatial-aware models that benefit farmers.

iv. Building farmer-owned data governance systems to prevent Big Tech from monopolising AI-driven farm insights.

v. Shaping global AI and digital agriculture policy in general to ensure fair, transparent, and equitable adoption of AI.

Spatial data engineering extends beyond agriculture to support broader societal objectives by promoting economic development, job creation, and informed governance. Improved agricultural productivity through spatial data-driven systems improves farm and agribusiness profitability, boosting rural economies. The advancement and application of spatial tools generate employment opportunities in the technology, data analysis, and advisory

sectors. Furthermore, policymakers can leverage spatial data for land use planning, designing targeted strategies, and ensuring compliance with sustainability standards. These initiatives are in harmony with global efforts, including the Sustainable Development Goals of the United Nations, which focus on zero hunger, sustainable production, and climate action.

6.4 Future research

Looking ahead, the future of spatial data engineering for digital agriculture depends not only on technological innovation but also on a deliberate and coordinated effort to improve connectivity across the socio-technical ecosystem. This work identifies four interrelated domains in which future research is especially critical: connecting data, connecting models, connecting systems, and connecting researchers. Each of these dimensions addresses current bottlenecks, ranging from fragmented datasets to isolated disciplinary practices, and outlines pathways toward more integrated, scalable, and human-centric digital agriculture. By advancing research in these areas, the digital agriculture community can better harness the full potential of geospatial data, artificial intelligence, and participatory innovation to support sustainable and equitable agricultural transitions.

6.4.1 *Data*: Enhancing interoperability and standardisation

A central challenge in the advancement of spatial data engineering for digital agriculture lies in connecting heterogeneous data sources in a meaningful, consistent, and interoperable manner. At present, data ecosystems in agriculture remain highly fragmented, with critical obstacles including: (i) fragmentation of data sources across sectors and stakeholders; (ii) mismatches in spatial and temporal resolution between datasets; and (iii) limited capabilities for real-time integration of multi-source data streams.

To overcome these issues, future research must prioritise the development of frameworks and tools that enable seamless data connectivity. Key directions include:

i. Global geospatial data standards for digital agriculture: Establishing open and widely adopted standards for geospatial data formats and metadata, tailored for use in digital agriculture, will enhance interoperability between platforms, projects, and regions.

ii. Automated spatial-temporal harmonisation: Advanced methods are needed to reconcile data collected at different resolutions or frequencies. Automated harmonisation techniques can improve consistency and facilitate integrated analyses across scales.

iii. Decentralised data processing through edge computing and federated learning: These approaches allow data to be processed locally–on-farm or at the sensor level–while preserving privacy and reducing bandwidth demands. They are especially important

in empowering farmers to retain control over their data while contributing to broader analytics.

iv. AI-driven feature extraction for geospatial analytics: Leveraging artificial intelligence to automatically identify relevant features from diverse datasets (e.g., satellite imagery, in-situ sensors, weather feeds) will support scalable, real-time modelling and enhance the utility of spatial data for decision support.

6.4.2 *Models*: Advancing AI, geospatial analytics, and digital twins

As digital agriculture continues to evolve, the integration and advancement of analytical models–particularly in geospatial and AI domains–remains a critical area for future research. Despite substantial progress in both physics-based and data-driven modelling approaches, several challenges persist that limit their full potential in real-world agricultural systems. These include: (i) a lack of integration between mechanistic crop models and AI-based geospatial models; (ii) limited operational deployment of real-time models for early warning systems (e.g., pest outbreaks, extreme weather); and (iii) difficulties in model generalisation due to the heterogeneity of landscapes, soils, and farming practices.

To address these limitations and move toward a more predictive, adaptive, and context-aware modelling ecosystem, the following research priorities are identified:

i. Development of hybrid AI-physics models: Combining process-based crop simulation models with deep learning techniques offers the potential to merge domain knowledge with data-driven flexibility. These hybrid models could provide improved robustness and scalability for precision agriculture applications.

ii. Self-adaptive (spatial-aware) models: Future models should be capable of learning and dynamically adapting to changing environmental conditions, cropping systems, and management practices, i.e. be more robust to data and concept drift. This adaptability is essential to maintain relevance in the face of climate variability and evolving agricultural trends.

iii. AI-based uncertainty quantification: Enhancing the transparency of the model through the estimation and communication of uncertainties is critical to trust and usability. New research is needed in applying probabilistic and ensemble AI techniques to quantify uncertainty in geospatial forecasting and remote sensing-based agricultural predictions.

iv. Geospatial Digital Twins for agriculture: Digital Twins–virtual replicas of real-world agro-ecosystems–offer a promising framework for real-time simulation, monitoring, and decision support. Research should explore scalable architectures for Digital Twins that integrate sensor data, remote sensing, and models to simulate agricultural dynamics in near real-time. The LTER-LIFE initiative might serve as an example.

6.4.3 *Systems*: Scaling geospatial pipelines and cloud-AI integration

The operationalisation of spatial data engineering at scale requires robust system architectures that can efficiently manage and process vast volumes of geospatial data. As digital agriculture increasingly relies on high-resolution imagery, sensor networks, and AI-driven analytics, system-level challenges related to computational load, latency, and energy efficiency are becoming more prominent. Current limitations include: (i) the high computational cost of processing complex geospatial datasets (e.g., hyper-spectral imaging, lidar); (ii) insufficient capabilities for real-time AI inference in edge and cloud computing environments; and (iii) the energy-intensive nature of training and deploying deep learning models for geospatial applications.

To overcome these bottlenecks and ensure scalable, efficient, and sustainable data infrastructures, future research should focus on the following directions:

i. Serverless geospatial computing: Using serverless architectures can facilitate cost-effective on-demand processing of agricultural data. This approach eliminates the need for continuous infrastructure provisioning, making geospatial analytics more accessible and scalable for diverse agricultural stakeholders.

ii. AI-powered edge computing: Deploying AI models directly on edge devices–such as IoT-enabled farm sensors–can significantly reduce data transfer latency and enable real-time responses to field conditions. This decentralised approach also supports greater data privacy and resilience in network-constrained environments.

iii. Energy-efficient geospatial AI: The environmental footprint of large-scale geospatial AI must be addressed through innovations such as neuromorphic computing, which mimics biological neural processing, and emerging paradigms like quantum computing (Pook et al., 2025). These approaches promise to reduce energy demands while maintaining high analytical performance.

iv. Decentralised geospatial infrastructure: Future systems should explore peer-to-peer data exchange networks that allow decentralised sharing and processing of agricultural data. This model improves resilience, reduces central bottlenecks, and empowers local actors to retain control over their data assets while contributing to broader knowledge ecosystems.

6.4.4 *Researchers*: Strengthening collaboration

The transformative potential of spatial data engineering in digital agriculture can only be fully realised through effective collaboration across disciplinary, institutional, and geographic boundaries. However, several critical barriers continue to limit this collaboration. These include: (i) a lack of structured interdisciplinary engagement among agronomists, data engineers, and AI researchers; (ii) unequal access to data and digital tools, particularly for smallholder farmers who often remain excluded from the benefits of advanced geospatial

analytics; and (iii) growing ethical concerns around the use of AI in agriculture, such as algorithmic bias in soil fertility models or decision support systems that do not reflect local farming realities.

To address these challenges and foster inclusive, responsible innovation in digital agriculture, future research should prioritise the following areas:

i. Open-access geospatial data hubs: Shared platforms for storing, accessing, and exchanging agricultural geospatial data can lower barriers to entry, encourage collaborative innovation, and support transparency. These centres should prioritise accessibility, metadata quality, and interoperability to serve both researchers and practitioners.

ii. Co-designed AI-based decision support systems: Active participation of farmers in the design and development of AI tools is essential to ensure usability, relevance, and trust. Co-design processes that bring together farmers, agronomists, and data scientists can help tailor digital tools to real-world agricultural needs and contexts.

iii. Ethical AI frameworks for agriculture: As AI becomes more embedded in geospatial decision-making, research must address fairness, accountability, and transparency. This includes ensuring that models are interpretable, avoiding unintended biases, and establishing governance structures that align with local values and regulations.

iv. Participatory research methodologies: Engaging local farmers, especially those from under-represented or resource-constrained communities, in the research and innovation process promotes both equity and adoption. Participatory approaches can help validate models against lived experience, surface tacit knowledge, and build stronger trust in digital systems.

6.4.5 *Connecting...*

Taken together, these four research avenues underscore a unifying principle of this thesis: that spatial data engineering must function as a connective tissue–linking not only digital assets and computational tools, but also disciplinary knowledge, human expertise, and on-the-ground realities. Connecting data ensures interoperability and coherence; connecting models brings together explanatory and predictive power; connecting systems enables scalable and energy-efficient analytics; and connecting researchers promotes inclusive, ethical innovation. Future efforts that embrace this integrative and collaborative ethos will be best positioned to support trustworthy, context-aware, and widely adopted digital solutions–ultimately contributing to resilient agricultural systems in an increasingly data-driven world.

6.5 Final thoughts

In The Wizard and the Prophet, Mann, 2018 contrasts two enduring paradigms for addressing environmental and agricultural challenges: one advocating technological innovation to expand resources (the Wizard) and the other emphasising ecological limits and the need for restraint (the Prophet). As a closing thought, here is how spatial data engineering can play a role in both of these pathways.

In a *technocratic scenario*, AI and machine learning have become deeply embedded in modern agricultural decision making. Lightweight semantic structures are streamlined to support real-time data fusion, predictive modelling, and cross-domain systems (e.g., integrating agri-food, climate, and economic models). Spatial data plays a central role in model calibration, model training, site-specific advisory systems, and adaptive policy feedback loops. Here, knowledge graphs and ontological metadata act as *semantic glue* linking data, models, and decision systems into coherent, connected, and dynamic infrastructures.

In a *post-tech scenario*, there is a societal shift away from large-scale AI, toward agroecological sovereignty and low-tech resilience. Spatial data infrastructures serve local communities, focussing on traditional knowledge, seed sovereignty, and participatory mapping. Semantics remain relevant, not for automation, but for *documenting and sharing place-based knowledge* using controlled vocabularies and open linked data. Farmer-centric approaches, developed and tested in close collaboration, where they keep full control over their data (e.g., it does not leave the farm), illustrate this trend toward decentralised, transparent, and human-centric spatial data use.

Figure 6.1: Autonomous farm robots, as envisioned in 2025 (Suwin66, 2025)

What unites both scenarios is the recognition that better connections between data, models, systems, and researchers *are essential and that spatial data engineering will continue to play an important foundational role in this for digital agriculture.*

❄

References

Afshar, M. H., T. Foster, T. P. Higginbottom, B. Parkes, K. Hufkens, S. Mansabdar, F. Ceballos, and B. Kramer (2021). "Improving the Performance of Index Insurance Using Crop Models and Phenological Monitoring". *Remote Sensing* 13.5. DOI: 10.3390/rs13050924.

Ahmed, I., M. S. Poole, and A. Trudeau (2018). "A Typology of Virtual Research Environments". In: *Proceedings of the 51st Hawaii International Conference on System Sciences (HICSS)*. Waikoloa Village, HI, USA: University of Hawaii at Manoa, 448–457. DOI: 10.24251/HICSS.2018.087.

Alderman, P. (2021). *Parallel gridded simulation framework for DSSAT-CSM (version 4.7.5.21) using MPI and NetCDF*. Copernicus GmbH. DOI: 10.5194/gmd-2021-183.

Allan, R. N. (2009). *Virtual Research Environments: From Portals to Science Gateways*. 1st. Chandos Information Professional Series. Oxford: Chandos Publishing (Woodhead Publishing/Elsevier imprint), 284.

Allen, I. and C. Seaman (2007). "Likert scales and data analyses". *Quality progress* 40.7, 64–65.

Altintas, I., C. Berkley, E. Jaeger, M. Jones, B. Ludäscher, and S. Mock (2004). "Kepler: An Extensible System for Design and Execution of Scientific Workflows". In: *Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM)*. Santorini Island, Greece: IEEE Computer Society, 423–424. DOI: 10.1109/SSDM.2004.1311241.

Amiri-Zarandi, M., M. Hazrati Fard, S. Yousefinaghani, M. Kaviani, and R. Dara (2022). "A Platform Approach to Smart Farm Information Processing". *Agriculture* 12.6, 838. DOI: 10.3390/agriculture12060838.

Argent, R. M. (2004). "An overview of model integration for environmental applications—components, frameworks and semantics". *Environmental Modelling & Software* 19.3, 219–234. DOI: 10.1016/S1364-8152(03)00150-6.

Argent, R. M., A. A. Voinov, T. Maxwell, S. M. Cuddy, J. M. Rahman, S. P. Seaton, R. A. Vertessy, and R. D. Braddock (2006). "Comparing modelling frameworks: a

workshop approach". *Environmental Modelling & Software* 21.7, 895–910. DOI: 10.1016/j.envsoft.2005.05.004.

Assante, M., A. Boizet, L. Candela, D. Castelli, R. Cirillo, G. Coro, E. Fernandez, M. Filter, L. Frosini, G. Kakaletris, P. Katsivelis, M. Knapen, L. Lelii, R. Lokers, F. Mangiacrapa, P. Pagano, G. Panichi, L. Penev, F. Sinibaldi, and P. Zervas (2020). "Realizing virtual research environments for the agri-food community: The AGINFRA PLUS experience". *Concurrency and Computation: Practice and Experience*.

Assante, M., L. Candela, D. Castelli, and R. Cirillo (2019). "Enacting Open Science by D4Science". *Future Generation Computer Systems* 101, 555–563. DOI: 10.1016/j.future.2019.05.063.

Athanasiadis, I. N. (2015). "Challenges in Modelling of Environmental Semantics". In: *Environmental Software Systems. Infrastructures, Services and Applications*. Ed. by R. Denzer, R. M. Argent, G. Schimak, and J. Hřebíček. Vol. 448. IFIP Advances in Information and Communication Technology. Cham: Springer International Publishing, 19–25. DOI: 10.1007/978-3-319-15994-2_2.

Athanasiadis, I. N., A.-E. Rizzoli, S. J. C. Janssen, E. Andersen, and F. Villa (2009). "Ontology for Seamless Integration of Agricultural Data and Models". In: *Metadata and Semantic Research (MTSR 2009), Communications in Computer and Information Science, vol. 46*. Ed. by F. Sartori, M. Sicilia, and N. Manouselis. Milan, Italy: Springer, Berlin & Heidelberg, 282–293. DOI: 10.1007/978-3-642-04590-5_27.

Backus, J. (1978). "Can programming be liberated from the von Neumann style?" *Commun. ACM* 21.8, 613–641.

Bampasidou, M., D. Goldgaber, T. Gentimis, and A. Mandalika (2024). "Overcoming 'Digital Divides': Leveraging higher education to develop next generation digital agriculture professionals". *Computers and Electronics in Agriculture* 224, 109181. DOI: 10.1016/j.compag.2024.109181.

Barbosa Júnior, M., B. de Almeida Moreira, V. Carreira, A. de Brito Filho, C. Trentin, F. de Souza, D. Tedesco, T. Setiyono, J. Flores, Y. Ampatzidis, R. da Silva, and L. Shiratsuchi (2024). "Precision agriculture in the United States: A comprehensive meta-review inspiring further research, innovation, and adoption". *Computers and Electronics in Agriculture* 221.

Barker, M., S. D. Olabarriaga, N. Wilkins-Diehr, S. Gesing, D. S. Katz, S. Shahand, S. Henwood, T. Glatard, K. Jeffery, and B. Corrie (2019). "The Global Impact of Science Gateways, Virtual Research Environments and Virtual Laboratories". *Future Generation Computer Systems* 95, 240–248. DOI: 10.1016/j.future.2018.12.026.

Bertoglio, L. R., M. Jarke, and J. Lässig (2021). "A Survey on Digital Agriculture: Research Trends, Challenges and Opportunities". *arXiv preprint* arXiv:2103.12488.

Boogaard, H. and G. V. der Grijn (2020). *Agrometeorological indicators from 1979 to present derived from reanalysis. Copernicus Climate Change Service*. ECMWF Copernicus. DOI: https://doi.org/10.24381/cds.6c68c9bb.

Boyd danah, d. and K. Crawford (2012). "Critical Questions for Big Data: Provocations for a Cultural, Technological, and Scholarly Phenomenon". *Information, Communication & Society* 15.5, 662–679. DOI: 10.1080/1369118X.2012.678878.

Brewer, E. (2012). "CAP twelve years later: How the "rules" have changed". *Computer* 45.2, 23–29.

Britz, W., I. P. Domínguez, and T. Heckelei (2009). "A Comparison of CAPRI and SEAMLESS-IF as Integrated Modelling Systems". In: *Environmental and Agricultural Modelling: Integrated Approaches for Policy Impact Assessment*. Ed. by F. Brouwer and M. K. Van Ittersum. Dordrecht: Springer, 257–274. DOI: 10.1007/978-90-481-3619-3_11.

Calyam, P., N. Wilkins-Diehr, M. Miller, E. H. Brookes, R. Arora, A. Chourasia, D. M. Jennewein, V. Nandigam, M. D. LaMar, S. B. Cleveland, G. Newman, S. Wang, I. Zaslavsky, M. A. Cianfrocco, K. Ellett, D. Tarboton, K. G. Jeffery, Z. Zhao, J. González-Aranda, M. J. Perri, G. Tucker, L. Candela, T. Kiss, and S. Gesing (2021). "Measuring Success for a Future Vision: Defining Impact in Science Gateways/Virtual Research Environments". *Concurrency and Computation: Practice and Experience* 33.19, e6235. DOI: 10.1002/cpe.6099.

Cao, L. (2018). *Data Science Thinking: The Next Scientific, Technological and Economic Revolution*. Data Analytics. Cham: Springer. DOI: 10.1007/978-3-319-95092-1.

Carberry, P., Z. Hochman, R. McCown, N. Dalgliesh, M. Foale, P. Poulton, J. Hargreaves, D. Hargreaves, S. Cawthray, N. Hillcoat, and M. Robertson (2002). "The FARMSCAPE approach to decision support: farmers', advisers', researchers' monitoring, simulation, communication and performance evaluation". *Agricultural Systems* 74.1, 141–177. DOI: 10.1016/S0308-521X(02)00025-2.

Carpenter, S. R., H. A. Mooney, J. Agard, D. Capistrano, R. S. DeFries, S. Díaz, T. Dietz, A. K. Duraiappah, A. Oteng-Yeboah, H. M. Pereira, C. Perrings, W. V. Reid, J. Sarukhán, R. J. Scholes, and A. Whyte (2009). "Science for Managing Ecosystem Services: Beyond the Millennium Ecosystem Assessment". *Proceedings of the National Academy of Sciences of the United States of America* 106.5, 1305–1312. DOI: 10.1073/pnas.0808772106.

Cash, D. W., W. C. Clark, F. Alcock, N. M. Dickson, N. Eckley, D. H. Guston, J. Jäger, and R. B. Mitchell (2003). "Knowledge systems for sustainable development". *Proceedings of the National Academy of Sciences of the United States of America* 100.14, 8086–8091. DOI: 10.1073/pnas.1231332100.

Ceccarelli, T., S. Kannan, F. Cecchi, and S. Janssen (2022). *Contributions of information and communication technologies to food systems transformation*. Tech. rep. 82. Rome,

Italy: International Fund for Agricultural Development (IFAD), 38. DOI: 10.22004/ag.econ.322003.

Chlingaryan, A., S. Sukkarieh, and B. M. Whelan (2018). "Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review". *Computers and Electronics in Agriculture* 151, 61–69. DOI: 10.1016/j.compag.2018.05.012.

David, O., S. Markstrom, K. Rojas, L. Ahuja, and M. Schneider (2002). "The Object Modeling System". In: *Agricultural System Models in Field Research and Technology Transfer*. Ed. by L. Ahuja and T. Howell. Boca Raton, FL, USA: Lewis Publishers, CRC Press LLC, 317–331.

De Wit, A. (2021). *PCSE: The Python Crop Simulation Environment*. https://pcse.readthedocs.io.

De Wit, A., H. Boogaard, I. Supit, and M. Van den Berg, eds. (2020a). *System description of the WOFOST 7.2, cropping systems model*. English. rev. 1.1. Wageningen Environmental Research.

– (2020b). *System description of the WOFOST 7.2, cropping systems model*. English. Tech. rep. Publisher: Wageningen Environmental Research. Wageningen Environmental Research.

De Wit, A., H. Boogaard, D. Fumagalli, S. J. C. Janssen, M. Knapen, D. Van Kraalingen, I. Supit, R. Van der Wijngaart, and K. Van Diepen (2019). "25 Years of the WOFOST Cropping Systems Model". *Agricultural Systems* 168, 154–167. DOI: 10.1016/j.agsy.2018.06.018.

De Wit, A., S. Hoek, R. Ballaghi, T. El Hairech, and D. Qinghan (2013). "Building an operational system for crop monitoring and yield forecasting in Morocco". In: *2013 Second International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, 466–469. DOI: 10.1109/Argo-Geoinformatics.2013.6621964.

Dean, J. and S. Ghemawat (2008). "MapReduce: Simplified Data Processing on Large Clusters". *Communications of the ACM* 51.1, 107–113. DOI: 10.1145/1327452.1327492.

Donatelli, M., G. Russell, A. E. Rizzoli, M. Acutis, M. Adam, I. N. Athanasiadis, M. Balderacchi, L. Bechini, H. Belhouchette, G. Bellocchi, J.-E. Bergez, M. Botta, N. Brisson, S. Bregaglio, L. Carlini, R. Confalonieri, L. Doro, F. Gerri, D. Giltrap, C. Grignani, C. Huyghe, M. Knapen, R. Kröbel, P. A. Leffelaar, R. Magarey, L. Manceau, G. Martin, ..., and M. K. Van Ittersum (2010). "A Component-Based Framework for Simulating Agricultural Production and Externalities". In: *Environmental and Agricultural Modelling: Integrated Approaches for Policy Impact Assessment*. Ed. by M. Bindi, G. Flichman, G. Trombi, M. Donatelli, D. Leclère, and P. Moutonnet. Dordrecht: Springer, 63–108. DOI: 10.1007/978-90-481-3619-3_4.

Donchyts, G., S. Hummel, S. Vaneček, J. Groos, A. Harper, M. Knapen, J. Gregersen, P. Schade, A. Antonello, and P. Gijsbers (2010). "OpenMI 2.0—What's New?" In:

Proceedings of the 5th Biennial Congress of the International Environmental Modelling and Software Society (iEMSs 2010): Modelling for Environment's Sake, Volume 2. Ed. by D. A. Swayne, W. Yang, A. A. Voinov, A. Rizzoli, and T. Filatova. International Environmental Modelling and Software Society (iEMSs). Ottawa, Ontario, Canada, 1100–1106.

EUDAT, LIBER, OpenAIRE, EGI, and GÉANT (2015). *Position Paper: European Open Science Cloud for Research*. Zenodo. DOI: 10.5281/zenodo.32915. URL: https://doi.org/10.5281/zenodo.32915.

Ewert, F., M. V. Ittersum, I. Bezlepkina, O. Therond, E. Andersen, H. Belhouchette, C. Bockstaller, F. M. Brouwer, T. Heckelei, S. Janssen, M. Knapen, M. Kuiper, K. Louhichi, J. A. Olsson, N. Turpin, J. Wery, J. Wien, and J. Wolf (2009). "A methodology for enhanced flexibility of integrated assessment in agriculture". *Environmental Science & Policy* 12.5, 546–561. DOI: 10.1016/j.envsci.2009.02.005.

Fielding, R. T. (2000). "Architectural Styles and the Design of Network-based Software Architectures". Ph.D. dissertation. Irvine, CA, USA: University of California, Irvine.

Flynn, M. J. (1966). "Very High-Speed Computing Systems". *Proceedings of the IEEE* 54.12, 1901–1909. DOI: 10.1109/PROC.1966.5273.

– (1972). "Some Computer Organizations and Their Effectiveness". *IEEE Transactions on Computers* C-21.9, 948–960. DOI: 10.1109/TC.1972.5009071.

Food and Agriculture Organization of the United Nations (FAO) (2002). *The State of Food Insecurity in the World 2002: Food Insecurity – When People Must Live with Hunger and Fear Starvation*. Rome, Italy, 36.

– (2019). *Digital Technologies in Agriculture and Rural Areas: Briefing Paper*. Rome, 6. DOI: 10.4060/ca4887en.

Food and Agriculture Organization of the United Nations (FAO) and International Fund for Agricultural Development (IFAD) and United Nations Children's Fund (UNICEF) and World Food Programme (WFP) and World Health Organization (WHO) (2022). *The State of Food Security and Nutrition in the World 2022: Repurposing Food and Agricultural Policies to Make Healthy Diets More Affordable*. Rome, Italy, xxiii, 231. DOI: 10.4060/cc0639en.

Fountas, S., B. Espejo-García, A. Kasimati, M. Gemtou, H. Panoutsopoulos, and E. Anastasiou (2024). "Agriculture 5.0: Cutting-Edge Technologies, Trends, and Challenges". *IT Professional* 26, 40–47.

Fountas, S., G. Carli, C. G. Sørensen, Z. Tsiropoulos, C. Cavalaris, A. Vatsanidou, B. Liakos, M. Canavari, J. Wiebensohn, and B. Tisseyre (2015). "Farm management information systems: Current situation and future perspectives". *Computers and Electronics in Agriculture* 115, 40–50. DOI: 10.1016/j.compag.2015.05.011.

Franzen, D. and D. Mulla (2016). "A history of precision agriculture". *Precision agriculture technology for crop farming*.

Fritz, S., L. See, J. C. L. Bayas, F. Waldner, D. Jacques, and ... (2019). "A comparison of global agricultural monitoring systems and current gaps". *Agricultural systems* 168, 258–272.

Gadzhev, G., I. Georgieva, K. Ganev, and V. Ivanov (2018). "Climate Applications in a Virtual Research Environment Platform". *Scalable Computing: Practice and Experience* 19.2, 107–118. DOI: 10.12694/scpe.v19i2.1347.

Gao, S., Y. Hu, and W. Li, eds. (2023). *Handbook of Geospatial Artificial Intelligence*. 1st. Boca Raton: CRC Press (Taylor & Francis), 448. DOI: 10.1201/9781003308423.

Garousi, V., M. Borg, and M. Oivo (2020). "Practical Relevance of Software Engineering Research: Synthesizing the Community's Voice". *Empirical Software Engineering* 25.3, 1687–1754. DOI: 10.1007/s10664-020-09803-0.

Gebbers, R. and V. Adamchuk (2010). "Precision agriculture and food security". *Science* 327.5967, 828–831.

Gijsbers, P., R. Moore, and C. Tindall (2002). "HarmonIT: Towards OMI, an Open Modelling Interface and Environment to harmonise European developments in water related simulation software". In: *Hydroinformatics 2002, Volume Two: Software Tools and Management Systems*. Ed. by I. Cluckie, D. Han, J. Davis, and S. Heslop. Vol. 2. London, UK: IWA Publishing, 1268–1275.

Godfray, H. C. J., J. R. Beddington, I. R. Crute, L. Haddad, D. Lawrence, J. F. Muir, J. Pretty, S. Robinson, S. M. Thomas, and C. Toulmin (2010). "Food Security: The Challenge of Feeding 9 Billion People". *Science* 327.5967, 812–818. DOI: 10.1126/science.1185383.

Goldacre, B., C. E. Morton, and N. J. DeVito (2019). "Why Researchers Should Share Their Analytic Code". *BMJ* 367, l6365. DOI: 10.1136/bmj.l6365.

Golmohammadi, J., Y. Xie, J. Gupta, M. Farhadloo, Y. Li, J. Cai, S. Detor, A. Roh, and S. Shekhar (2020). "An Introduction to Spatial Data Mining". *Geographic Information Science & Technology Body of Knowledge* 2020.Q4, 5. DOI: 10.22224/gistbok/2020.4.5.

Goodall, J. L., B. F. Robinson, and A. M. Castronova (2011). "Modeling Water Resource Systems Using a Service-Oriented Computing Paradigm". *Environmental Modelling & Software* 26.5, 573–582. DOI: 10.1016/j.envsoft.2010.11.013.

Gregersen, J., P. Gijsbers, and S. Westen (2007). "OpenMI: Open modelling interface". *Journal of Hydroinformatics* 9.3, 175–191. DOI: 10.2166/hydro.2007.023.

Gruber, T. R. (1995). "Toward Principles for the Design of Ontologies Used for Knowledge Sharing". *International Journal of Human-Computer Studies* 43.5–6, 907–928. DOI: 10.1006/ijhc.1995.1081.

Guttman, A. (1984). "R-Trees: A Dynamic Index Structure for Spatial Searching". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, 47–57. DOI: 10.1145/602259.602266.

Hack-ten Broeke, M., H. Mulder, R. Bartholomeus, J. Van Dam, G. Holshof, I. Hoving, D. Walvoort, M. Heinen, J. Kroes, P. Van Bakel, I. Supit, A. De Wit, and R. Ruijtenberg (2019). "Quantitative land evaluation implemented in Dutch water management". *Geoderma* 338, 536–545. DOI: https://doi.org/10.1016/j.geoderma.2018.11.002.

Hamdani, Y., G. Xiao, L. Ding, and D. Calvanese (2023). "An Ontology-Based Framework for Geospatial Integration and Querying of Raster Data Cube Using Virtual Knowledge Graphs". *ISPRS International Journal of Geo-Information* 12.9, 375. DOI: 10.3390/ijgi12090375.

Hargitai, H., J. Wang, P. J. Stooke, I. Karachevtseva, Á. Kereszturi, and M. Gede (2017). "Map Projections in Planetary Cartography". In: *Choosing a Map Projection*. Ed. by M. Lapaine and E. L. Usery. Lecture Notes in Geoinformation and Cartography. Cham: Springer International Publishing, 177–202. DOI: 10.1007/978-3-319-51835-0_7.

Harris, G. (2002). "Integrated assessment and modelling: an essential way of doing science". *Environmental Modelling & Software* 17.3, 201–207. DOI: 10.1016/S1364-8152(01)00058-5.

Helming, K., M. Pérez-Soba, and P. Tabbush, eds. (2008). *Sustainability Impact Assessment of Land Use Changes*. Berlin, Heidelberg: Springer, x, 508. DOI: 10.1007/978-3-540-78648-1.

Hennessy, J. L. and D. A. Patterson (2011). *Computer Architecture, Fifth Edition: A Quantitative Approach*. 5th. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Hertel, T. W. and M. E. Tsigas (1997). "Structure of GTAP". In: *Global Trade Analysis: Modeling and Applications*. Ed. by T. W. Hertel. Cambridge: Cambridge University Press, 13–73. DOI: 10.1017/CBO9781139174688.003.

Hillyer, C., J. Bolte, F. V. Evert, and A. Lamaker (2003). "The ModCom modular simulation system". *European Journal of Agronomy* 18.3–4, 333–343. DOI: 10.1016/S1161-0301(02)00111-9.

Hinkel, J. (2008). "Transdisciplinary Knowledge Integration: Cases from Integrated Assessment and Vulnerability Assessment". WUR PhD thesis no. 4403. PhD thesis. Wageningen, The Netherlands: Wageningen University, 176. DOI: 10.18174/121973.

Höhl, A., I. Obadic, M.-Á. Fernández-Torres, H. Najjar, D. A. B. Oliveira, Z. Akata, A. Dengel, and X. X. Zhu (2024). "Opening the Black-Box: A Systematic Review on Explainable AI in Remote Sensing". *IEEE Geoscience and Remote Sensing Magazine* 12.4, 261–304. DOI: 10.1109/MGRS.2024.3467001.

Holzinger, A., I. Fister, I. Fister, H.-P. Kaul, and S. Asseng (2024). "Human-Centered AI in Smart Farming: Toward Agriculture 5.0". *IEEE Access* 12, 62199–62214. DOI: 10.1109/ACCESS.2024.3395532.

Horrocks, I. (2005). "OWL: A Description Logic Based Ontology Language". In: *Principles and Practice of Constraint Programming - CP 2005*. Ed. by P. Van Beek. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 5–8. DOI: 10.1007/11564751_2.

Hosseinzadeh, M., E. Azhir, O. H. Ahmed, M. Y. Ghafour, S. H. Ahmed, A. M. Rahmani, and B. Vo (2023). "Data Cleansing Mechanisms and Approaches for Big Data Analytics: A Systematic Study". *Journal of Ambient Intelligence and Humanized Computing* 14.1, 99–111. DOI: 10.1007/s12652-021-03590-2.

Hu, Y. and W. Li (2017). "Spatial Data Infrastructures". *arXiv preprint* abs/1707.03969. Also published in GIST BoK (Geographic Information Science & Technology Body of Knowledge). DOI: 10.22224/gistbok/2017.2.1.

Huang, J., J. L. Gómez-Dans, H. Huang, H. Ma, Q. Wu, P. E. Lewis, S. Liang, Z. Chen, J.-H. Xue, Y. Wu, F. Zhao, J. Wang, and X. Xie (2019). "Assimilation of remote sensing into crop growth models: Current status and perspectives". *Agricultural and Forest Meteorology* 276-277, 107609.

Huang, Y., Z.-x. CHEN, T. YU, X.-z. HUANG, and X.-f. GU (2018). "Agricultural remote sensing big data: Management and applications". *Journal of Integrative Agriculture* 17.9, 1915–1931. DOI: https://doi.org/10.1016/S2095-3119(17)61859-8.

Ingram, J., D. Maye, C. Bailye, A. Barnes, C. Bear, M. Bell, D. Cutress, L. Davies, A. De Boon, and L. Dinnie (2022). "What are the priority research questions for digital agriculture". *Land Use Policy* 114.

INSPIRE Temporary MIWP 2021-2024 sub-group 2.3.1 (2024). *D2.8.III.9 Data Specification on Agricultural and Aquaculture Facilities – Technical Guidelines*. Tech. rep. INSPIRE Technical Guidance. European Commission.

Jagers, H. (2010). "Linking Data, Models and Tools: An Overview". In: *Proceedings of the 5th International Congress on Environmental Modelling and Software (iEMSs), Ottawa, Ontario, Canada, July 2010*. Paper no. 101, available via iEMSs conference proceedings. Ottawa, Ontario, Canada.

Jakeman, A. J. and R. A. Letcher (2003). "Integrated assessment and modelling: features, principles and examples for catchment management". *Environmental Modelling & Software* 18.6, 491–501. DOI: 10.1016/S1364-8152(03)00024-0.

Jakobsen, C. H. and W. J. McLaughlin (2004). "Communication in Ecosystem Management: A Case Study of Cross-Disciplinary Integration in the Assessment Phase of the Interior Columbia Basin Ecosystem Management Project". *Environmental Management* 33.5, 591–605. DOI: 10.1007/s00267-003-2900-2.

Jang, W. S., Y. Lee, J. C. Neff, Y. Im, S. Ha, and L. Doro (2019). "Development of an EPIC parallel computing framework to facilitate regional/global gridded crop modeling with multiple scenarios: A case study of the United States". en. *Computers and Electronics in Agriculture* 158, 189–200. DOI: 10.1016/j.compag.2019.02.004.

Janssen, H., S. Janssen, K. MJR, W. Meijninger, Y. Van Randen, and I. la Riviere (2018). *AgroDataCube: A Big Open Data collection for Agri-Food Applications*.

Janssen, S., I. Athanasiadis, I. Bezlepkina, M. Knapen, H. Li, I. Domínguez, A. Rizzoli, and M. Van Ittersum (2011). "Linking Models for Assessing Agricultural Land Use Change". *Computers and Electronics in Agriculture* 76.2, 148–160. DOI: 10.1016/j.compag.2010.10.011.

Janssen, S. J. C. and M. K. Van Ittersum (2007). "Assessing Farm Innovations and Responses to Policies: A Review of Bio-Economic Farm Models". *Agricultural Systems* 94.3, 622–636. DOI: 10.1016/j.agsy.2007.03.001.

Janssen, S., C. Porter, A. Moore, I. Athanasiadis, I. Foster, J. Jones, and J. Antle (2017). "Towards a new generation of agricultural system data, models and knowledge products: Information and communication technology." *Agricultural Systems* 155, 200–212.

Jansson, T., M. Bakker, B. Boitier, A. Fougeyrollas, J. Helming, H. Van Meijl, and P. Verkerk (2008). "Cross sector land use modelling framework". In: *Sustainability Impact Assessment of Land Use Changes*. Ed. by K. Helming, M. Pérez-Soba, and P. Tabbush. Berlin–Heidelberg: Springer, 159–180. DOI: 10.1007/978-3-540-78648-1_9.

Kamilaris, A., A. Kartakoullis, and F. Prenafeta-Boldú (2017). "A review on the practice of big data analysis in agriculture". *Computers and Electronics in Agriculture* 143, 23–37.

Kamilaris, A. and F. X. Prenafeta-Boldú (2018). "Deep Learning in Agriculture: A Survey". *Computers and Electronics in Agriculture* 147, 70–90. DOI: 10.1016/j.compag.2018.02.016.

Karunathilake, E. M. B. M., A. T. Le, S. Heo, Y. S. Chung, and S. Mansoor (2023). "The Path to Smart Farming: Innovations and Opportunities in Precision Agriculture". *Agriculture* 13.8, 1593. DOI: 10.3390/agriculture13081593.

Kim, J., J. Y. Park, S. Hyun, B. H. Yoo, D. H. Fleisher, and K. S. Kim (2021). "Development of an orchestration aid system for gridded crop growth simulations using Kubernetes". *Computers and Electronics in Agriculture* 186, 106187. DOI: https://doi.org/10.1016/j.compag.2021.106187.

Kim, J., J. Park, S. Hyun, D. H. Fleisher, and K. S. Kim (2020). "Development of an automated gridded crop growth simulation support system for distributed computing with virtual machines". *Computers and Electronics in Agriculture* 169, 105196. DOI: https://doi.org/10.1016/j.compag.2019.105196.

Knapen, M., A. De Wit, E. Buyukkaya, P. Petrou, D. Paudel, S. Janssen, and I. Athanasiadis (2025). "Efficient and scalable crop growth simulations using standard big

data and distributed computing technologies". *Computers and Electronics in Agriculture* 236, 110392. DOI: https://doi.org/10.1016/j.compag.2025.110392.

Knapen, M., S. Janssen, O. Roosenschoon, P. Verweij, W. De Winter, M. Uiterwijk, and J.-E. Wien (2013). "Evaluating OpenMI as a model integration platform across disciplines". *Environmental Modelling & Software* 39, 274–282. DOI: https://doi.org/10.1016/j.envsoft.2012.06.011.

Knapen, M., R. Lokers, and S. Janssen (2023). "Evaluating the D4Science virtual research environment platform for agro-climatic research". *Agricultural Systems* 210, 103706. DOI: https://doi.org/10.1016/j.agsy.2023.103706.

Knapen, M., R. M. Lokers, L. Candela, and S. J. C. Janssen (2020). "AGINFRA PLUS: Running Crop Simulations on the D4Science Distributed e-Infrastructure". In: *Environmental Software Systems. Data Science in Action*. Ed. by J. Arias, C. Jung, L. M. Hilty, B. Penzenstadler, R. Chitchyan, and S. Reeves. Vol. 554. IFIP Advances in Information and Communication Technology. Cham, Switzerland: Springer International Publishing, 81–89. DOI: 10.1007/978-3-030-39815-6_8.

Knapen, M., P. Verweij, and J. Wien (2007). "Applying Enterprise Application Architectures in Integrated Modelling". In: *MODSIM 2007 International Congress on Modelling and Simulation*. Ed. by L. Oxley and D. Kulasiri. Modelling, Simulation Society of Australia, and New Zealand. Christchurch, New Zealand, 798–804.

Knapen, M., P. J. F. M. Verweij, and S. Janssen (2010). "Agilists and the Art of Integrated Assessment Tool Development". In: *Proceedings of the 2010 International Congress on Environmental Modelling and Software (iEMSs 2010): Modelling for Environment's Sake*. Ed. by D. A. Swayne, W. Yang, A. A. Voinov, A. Rizzoli, and T. Filatova. Vol. 3. Ottawa, Canada: International Environmental Modelling and Software Society (iEMSs), 1894–1901.

Knuth, D. E. (1992). *Literate Programming*. Vol. 27. CSLI Lecture Notes. Stanford, California: Center for the Study of Language and Information (CSLI).

Kramer, K., G. M. Hengeveld, M.-J. Schelhaas, B. Van der Werf, and W. De Winter (2013). "Genetic Adaptive Response: Missing Issue in Climate Change Assessment Studies". In: *Impacts World 2013: International Conference on Climate Change Effects*. Potsdam, Germany: Potsdam Institute for Climate Impact Research (PIK), 366–373.

Kurtzer, G. M., V. Sochat, and M. W. Bauer (2017). "Singularity: Scientific containers for mobility of compute." *PLoS One* 12.5, e0177459.

Lahlou, M. (2018). *CGMS-Maroc: National System for Agrometeorological monitoring*. U.S. Department of Agriculture. (Visited on 2022).

Laney, D. (2001). *3D Data Management: Controlling Data Volume, Velocity, and Variety*. Research Note (Application Delivery Strategies) ADS 6 Feb 01.949. META Group Research Note. META Group (subsequently Gartner).

Lecerf, R., A. Ceglar, R. López-Lozano, M. Van Der Velde, and B. Baruth (2019). "Assessing the information in crop model and meteorological indicators to forecast crop yield over Europe". *Agricultural Systems* 168, 191–202. DOI: https://doi.org/10.1016/j.agsy.2018.03.002.

Li, D., S. Wang, and D. Li (2015). *Spatial Data Mining: Theory and Application*. 1st ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 308. DOI: 10.1007/978-3-662-48538-5.

Li, Y., Y. Xie, and S. Shekhar (2023a). "Spatial Data Science". In: *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook*. Ed. by L. Rokach, O. Maimon, and E. Shmueli. 3rd ed. Cham, Switzerland: Springer, 401–422. DOI: 10.1007/978-3-031-24628-9_18.

Li, Z., Z. Qi, Y. Liu, Y. Zheng, and Y. Yang (2023b). "A modularized parallel distributed High–Performance computing framework for simulating seasonal frost dynamics in Canadian croplands". *Computers and Electronics in Agriculture* 212, 108057. DOI: https://doi.org/10.1016/j.compag.2023.108057.

Liakos, K. G., P. Busato, D. Moshou, S. Pearson, and D. Bochtis (2018). "Machine Learning in Agriculture: A Review". *Sensors* 18.8, 2674. DOI: 10.3390/s18082674.

Lindner, M., T. Suominen, T. Palosuo, J. García-Gonzalo, P. Verweij, S. Zudin, and R. Päivinen (2010). "ToSIA—A Tool for Sustainability Impact Assessment of Forest-Wood-Chains". *Ecological Modelling* 221.18, 2197–2205. DOI: 10.1016/j.ecolmodel.2009.08.006.

Liu, J., Q. Koziol, G. F. Butler, N. Fortner, M. Chaarawi, H. Tang, S. Byna, G. K. Lockwood, R. Cheema, K. A. Kallback-Rose, D. Hazen, and M. Prabhat (2018). "Evaluation of HPC Application I/O on Object Storage Systems". In: *2018 IEEE/ACM 3rd International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems (PDSW-DISCS)*. IEEE.

Liu, Y., H. V. Gupta, E. Springer, and T. Wagener (2008). "Linking Science with Environmental Decision Making: Experiences from an Integrated Modeling Approach to Supporting Sustainable Water Resources Management". *Environmental Modelling & Software* 23.7, 846–858. DOI: 10.1016/j.envsoft.2007.10.007.

Lloyd, W., O. David, J. C. A. II, K. W. Rojas, J. R. Carlson, G. H. Leavesley, P. Krause, T. R. Green, and L. R. Ahuja (2011). "Environmental Modeling Framework Invasiveness: Analysis and Implications". *Environmental Modelling & Software* 26.10, 1240–1250. DOI: 10.1016/j.envsoft.2011.03.011.

Lokers, R., M. Knapen, S. Janssen, Y. Van Randen, and J. Jansen (2016a). "Analysis of Big Data technologies for use in agro-environmental science". *Environmental Modelling & Software* 84, 494–504.

Lokers, R. and M. Knapen (2018). *AGINFRA PLUS D5.4: Agro-climatic and Economic Modelling Pilot Evaluation Report*. Tech. rep. Zenodo (OpenAIRE). DOI: 10.5281/zenodo.1311603.

Lokers, R., M. Knapen, S. Janssen, Y. Van Randen, and J. Jansen (2016b). "Analysis of Big Data technologies for use in agro-environmental science". *Environmental Modelling & Software* 84, 494–504. DOI: https://doi.org/10.1016/j.envsoft.2016.07.017.

Lokers, R., Y. Van Randen, M. Knapen, S. Gaubitzer, S. Zudin, and S. J. C. Janssen (2015). "Improving Access to Big Data in Agriculture and Forestry Using Semantic Technologies". In: *Metadata and Semantics Research*. Vol. 544. Communications in Computer and Information Science. Proceedings of the 9th Metadata and Semantics Research Conference (MTSR 2015). Cham: Springer International Publishing, 369–380. DOI: 10.1007/978-3-319-24129-6_32.

Lokers, R. M., S. Konstantopoulos, A. Stellato, M. Knapen, and S. J. C. Janssen (2014). "Designing Innovative Linked Open Data and Semantic Technologies in Agro-Environmental Modelling". In: *Proceedings of the 7th International Congress on Environmental Modelling and Software (iEMSs)*. Oral presentation — Stream A6: Semantics, Metadata and Ontologies of Natural Systems. San Diego, CA, USA.

Louhichi, K., A. Kanellopoulos, S. Janssen, G. Flichman, M. Blanco, H. Hengsdijk, T. Heckelei, P. Berentsen, A. Oude Lansink, and M. K. Van Ittersum (2010). "FSSIM, a Bio-Economic Farm Model for Simulating the Response of EU Farming Systems to Agricultural and Environmental Policies". *Agricultural Systems* 103.8, 585–597. DOI: 10.1016/j.agsy.2010.06.006.

Maier, M. W. (1998). "Architecting principles for systems-of-systems". *Systems Engineering* 1.4, 267–284. DOI: 10.1002/(SICI)1520-6858(1998)1:4<267::AID-SYS3>3.0.CO;2-D.

Mann, C. C. (2018). *The Wizard and the Prophet: Two Remarkable Scientists and Their Dueling Visions to Shape Tomorrow's World*. New York: Knopf Doubleday Publishing Group, 640.

Manyika, J., M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. Hung Byers (2011). *Big Data: The Next Frontier for Innovation, Competition, and Productivity*. Tech. rep. Executive summary available online. San Francisco, CA, USA: McKinsey Global Institute, 156. DOI: 10.2139/ssrn.2086394.

McAfee, A. and E. Brynjolfsson (2012). "Big Data: The Management Revolution". *Harvard Business Review* 90.10. Magazine article, 60–68.

Micha, R., J. L. Penalvo, F. Cudhea, F. Imamura, C. D. Rehm, and D. Mozaffarian (2017). "Association Between Dietary Factors and Mortality From Heart Disease, Stroke, and Type 2 Diabetes in the United States". *JAMA* 317.9, 912–924. DOI: 10.1001/jama.2017.0947.

Miller, G. A. (1995). "WordNet: A Lexical Database for English". *Communications of the ACM* 38.11, 39–41. DOI: 10.1145/219717.219748.

Moore, A., D. Holzworth, N. Herrmann, N. Huth, and M. Robertson (2007). "The Common Modelling Protocol: A hierarchical framework for simulation of agricultural and

environmental systems". *Agricultural Systems* 95.1–3, 37–48. DOI: 10.1016/j.agsy.2007.03.006.

Moore, R. V. and C. I. Tindall (2005). "An overview of the Open Modelling Interface and Environment (the OpenMI)". *Environmental Science & Policy* 8.3, 279–286. DOI: doi.org/10.1016/j.envsci.2005.03.009.

Moran, D. D., M. Wackernagel, J. A. Kitzes, S. H. Goldfinger, and A. Boutaud (2008). "Measuring sustainable development—Nation by nation". *Ecological economics* 64.3, 470–474.

Nabuurs, G., M. Schelhaas, and A. Pussinen (2000). "Validation of the European forest information scenario model (EFISCEN) and a projection of Finnish forests". English. *Silva Fennica* 51.2/3, 167–179.

Nature Editorial Board (2009). "Data's shameful neglect". *Nature* 461.7261. Editorial, 145. DOI: 10.1038/461145a.

– (2018). "Does your code stand up to scrutiny?" *Nature* 555.7695. Editorial, 142. DOI: 10.1038/d41586-018-02741-4.

Networked European Software and Services Initiative (NESSI) (2012). *Big Data: A New World of Opportunities*. White Paper. Contribution of the European Technology Platform NESSI to the Big Data research and innovation discourse. Europe: NESSI (Networked European Software and Services Initiative), 1–25.

Ng, H. T. and J. M. Zelle (1997). "Corpus-Based Approaches to Semantic Interpretation in NLP". *AI Magazine* 18.4, 45–64. DOI: 10.1609/aimag.v18i4.1329.

Obi Reddy, G. P., B. S. Dwivedi, and G. Ravindra Chary (2023). "Applications of Geospatial and Big Data Technologies in Smart Farming". In: *Smart Agriculture for Developing Nations: Advanced Technologies and Societal Change*. Ed. by K. Pakeerathan. Advanced Technologies and Societal Change. Singapore: Springer Nature Singapore, 15–31. DOI: 10.1007/978-981-19-8738-0_2.

Open Geospatial Consortium (2018). *OpenGIS Geography Markup Language (GML) Encoding Standard Version 3.2.2*. Tech. rep. OGC 07-036r1. OGC Standard Document. Open Geospatial Consortium.

Osrof, H. Y., C. L. Tan, G. Angappa, S. F. Yeo, and K. H. Tan (2023). "Adoption of Smart Farming Technologies in Field Operations: A Systematic Review and Future Research Agenda". *Technology in Society* 75, 102400. DOI: 10.1016/j.techsoc.2023.102400.

Paracchini, M. L., C. Pacini, M. L. M. Jones, and M. Pérez-Soba (2011). "An Aggregation Framework to Link Indicators Associated with Multifunctional Land Use to the Stakeholder Evaluation of Policy Options". *Ecological Indicators* 11.1, 71–80. DOI: 10.1016/j.ecolind.2009.04.006.

Parra-López, C., S. Abdallah, G. Garcia-Garcia, A. Hassoun, P. Sánchez-Zamora, H. Trollman, S. Jagtap, and C. Carmona-Torres (2024). "Integrating digital technologies in

agriculture for climate change adaptation and mitigation: State of the art and future perspectives". *Computers and Electronics in Agriculture* 226.

Paudel, B., S. Riaz, S. W. Teng, R. R. Kolluri, and H. Sandhu (2025). "The Digital Future of Farming: A Bibliometric Analysis of Big Data in Smart Farming Research". *Cleaner and Circular Bioeconomy* 10, 100132. DOI: 10.1016/j.clcb.2024.100132.

Paudel, D., A. De Wit, H. Boogaard, D. Marcos, S. Osinga, and I. N. Athanasiadis (2023). "Interpretability of deep learning models for crop yield forecasting". *Computers and Electronics in Agriculture* 206, 107663. DOI: 10.1016/j.compag.2023.107663.

Pérez-Soba, M., S. Petit, L. Jones, N. Bertrand, V. Briquel, L. Omodei-Zorini, C. Contini, K. Helming, J. H. Farrington, M. T. Mossello, D. Wascher, F. Kienast, and R. D. Groot (2008). "Land Use Functions—A Multifunctionality Approach to Assess the Impact of Land Use Changes on Land Use Sustainability". In: *Sustainability Impact Assessment of Land Use Changes*. Ed. by K. Helming, M. Pérez-Soba, and P. Tabbush. Berlin, Heidelberg: Springer, 375–404. DOI: 10.1007/978-3-540-78648-1_19.

Phillips, P. W. B., J.-A. Relf-Eckstein, G. Jobe, and B. Wixted (2019). "Configuring the New Digital Landscape in Western Canadian Agriculture". *NJAS: Wageningen Journal of Life Sciences* 90–91, 100295. DOI: 10.1016/j.njas.2019.04.001.

Poggio, L., L. M. de Sousa, N. H. Batjes, G. B. M. Heuvelink, B. Kempen, E. Ribeiro, and D. Rossiter (2021). "SoilGrids 2.0: producing soil information for the globe with quantified spatial uncertainty". *SOIL* 7.1, 217–240. DOI: 10.5194/soil-7-217-2021.

Pook, T., J. Vandenplas, J. C. Boschero, E. Aguilera, K. Leijnse, A. Chauhan, Y. Bouzembrak, M. Knapen, and M. Aldridge (2025). "Assessing the Potential of Quantum Computing in Agriculture". *Computers and Electronics in Agriculture* 235, 110332. DOI: 10.1016/j.compag.2025.110332.

Porter, C. H., C. Villalobos, D. Holzworth, R. Nelson, J. W. White, I. N. Athanasiadis, S. Janssen, D. Ripoche, J. Cufi, D. Raes, M. Zhang, M. Knapen, R. Sahajpal, K. Boote, and J. W. Jones (2014). "Harmonization and Translation of Crop Modeling Data to Ensure Interoperability". *Environmental Modelling & Software* 62, 495–508. DOI: 10.1016/j.envsoft.2014.09.004.

Potschin, M. and R. Haines-Young (2008). "Sustainability Impact Assessments: limits, thresholds and the Sustainability Choice Space". In: *Sustainability Impact Assessment of Land Use Changes*. Ed. by K. Helming, M. Pérez-Soba, and P. Tabbush. Berlin, Heidelberg: Springer, 425–450. DOI: 10.1007/978-3-540-78648-1_21.

Rahman, J. M., S. P. Seaton, J.-M. Perraud, H. Hotham, D. I. Verrelli, and J. R. Coleman (2003). "It's TIME for a New Environmental Modelling Framework". In: *Proceedings of MODSIM 2003 International Congress on Modelling and Simulation, Volume 4*. Ed. by D. A. Post. Modelling, Simulation Society of Australia, and New Zealand, Inc. Townsville, Australia, 1727–1732.

Reis, J. and M. Housley (2022). *Fundamentals of Data Engineering: Plan and Build Robust Data Systems*. 1st. Sebastopol, CA: O'Reilly Media, 447.

Rijgersberg, H., M. Wigham, and J. L. Top (2011). "How semantics can improve engineering processes: A case of units of measure and quantities". *Advanced Engineering Informatics* 25.2, 276–287. DOI: 10.1016/j.aei.2010.07.008.

Rizzoli, A. E., M. Donatelli, I. N. Athanasiadis, F. Villa, and D. Huber (2008). "Semantic links in integrated modelling frameworks". *Mathematics and Computers in Simulation* 78.2–3, 412–423. DOI: 10.1016/j.matcom.2008.01.017.

Roscher, R., L. Roth, C. Stachniss, and A. Walter (2023). "Data-Centric Digital Agriculture: A Perspective". *arXiv preprint* abs/2312.03437. DOI: 10.48550/arXiv.2312.03437.

Ross, K. W., M. E. Brown, J. P. Verdin, and L. W. Underwood (2009). "Review of FEWS NET biophysical monitoring requirements". *Environmental Research Letters* 4.2, 024009. DOI: 10.1088/1748-9326/4/2/024009.

Rosser, B. (1941). "Alonzo Church. The calculi of lambda-conversion. Annals of Mathematics studies, no. 6. Lithoprinted. Princeton University Press, Princeton1941, 77 pp." *The Journal of Symbolic Logic* 6.4, 171–172. DOI: 10.2307/2267126.

Rotz, S., E. Duncan, M. Small, J. Botschner, R. Dara, I. Mosby, M. Reed, and E. D. G. Fraser (2019). "The Politics of Digital Agricultural Technologies: A Preliminary Review". *Sociologia Ruralis* 59.2, 203–229. DOI: 10.1111/soru.12233.

Rowley, J. (2007). "The Wisdom Hierarchy: Representations of the DIKW Hierarchy". *Journal of Information Science* 33.2, 163–180. DOI: 10.1177/0165551506070706.

Roy, P. V. and S. Haridi (2004). *Concepts, Techniques, and Models of Computer Programming*. MIT Press.

Ruiz-Real, J. L., J. Uribe-Toril, J. A. Torres Arriaza, and J. de Pablo Valenciano (2020). "A Look at the Past, Present and Future Research Trends of Artificial Intelligence in Agriculture". *Agronomy* 10.11, 1839. DOI: 10.3390/agronomy10111839.

Samourkasidis, A. and I. N. Athanasiadis (2020). "A semantic approach for timeseries data fusion". *Computers and Electronics in Agriculture* 169, 105171. DOI: 10.1016/j.compag.2019.105171.

San Emeterio de la Parte, M., J.-F. Martínez-Ortega, P. Castillejo, and N. Lucas Martínez (2024). "Spatio-temporal semantic data management systems for IoT in agriculture 5.0: Challenges and future directions". *Internet of Things* 25, 101030. DOI: 10.1016/j.iot.2023.101030.

San Emeterio de la Parte, M., J.-F. Martínez-Ortega, V. Hernández Díaz, and N. Lucas Martínez (2023). "Big Data and precision agriculture: a novel spatio-temporal semantic IoT data management framework for improved interoperability". *Journal of Big Data* 10, 52. DOI: 10.1186/s40537-023-00729-0.

Schade, P., G. Lang, and J. Jürges (2008). "OpenMI Compliant Import of Initial and Boundary Data into a Numerical 3D Model". In: *Proceedings of the iEMSs 2008 International Congress on Environmental Modelling and Software (iEMSs 2008), Volume 2*. Ed. by M. Sànchez-Marrè, J. Béjar, J. Comas, A. Rizzoli, and G. Guariso. International Environmental Modelling and Software Society (iEMSs). Barcelona, Catalonia, Spain, 1086–1093.

Schieffer, J. and C. R. Dillon (2015). "The economic and environmental impacts of precision agriculture and interactions with agro-environmental policy". *Precision Agriculture* 16.1, 46–61. DOI: 10.1007/s11119-014-9382-5.

Schmolke, A., P. Thorbek, D. L. DeAngelis, and V. Grimm (2010). "Ecological Models Supporting Environmental Decision Making: A Strategy for the Future". *Trends in Ecology & Evolution* 25.8, 479–486. DOI: 10.1016/j.tree.2010.05.001.

Scholten, H. (2008). "Better Modelling Practice: An Ontological Perspective on Multi-disciplinary, Model-Based Problem Solving". WUR PhD thesis no. 4562. PhD thesis. Wageningen, The Netherlands: Wageningen University, 314. DOI: 10.18174/122078.

Schwab, K. (2017). *The Fourth Industrial Revolution*. New York: Crown Business, 192.

Sharma, A., A. Jain, P. Gupta, and V. S. Chowdary (2021). "Machine Learning Applications for Precision Agriculture: A Comprehensive Review". *IEEE Access* 9, 4843–4873. DOI: 10.1109/ACCESS.2020.3048415.

Shepherd, M., J. A. Turner, B. Small, and D. Wheeler (2020). "Priorities for science to overcome hurdles thwarting the full promise of the 'digital agriculture' revolution". *Journal of the Science of Food and Agriculture* 100.14, 5083–5092. DOI: 10.1002/jsfa.9346.

Siddiqa, A., A. Karim, and A. Gani (2017). "Big data storage technologies: a survey". *Frontiers of Information Technology & Electronic Engineering* 18.8, 1040–1070.

Slavin, P. (2016). "Climate and famines: a historical reassessment". *Wiley Interdisciplinary Reviews: Climate Change* 7.3, 433–447. DOI: 10.1002/wcc.395.

Steffen, W., K. Richardson, J. Rockström, S. E. Cornell, I. Fetzer, E. M. Bennett, R. Biggs, S. R. Carpenter, W. De Vries, C. A. De Wit, C. Folke, D. Gerten, J. Heinke, G. M. Mace, L. M. Persson, V. Ramanathan, B. Reyers, and S. Sörlin (2015). "Planetary Boundaries: Guiding Human Development on a Changing Planet". *Science* 347.6223, 1259855. DOI: 10.1126/science.1259855.

Stoate, C., A. Baldi, P. Beja, N. Boatman, I. Herzon, A. Van Doorn, G. De Snoo, L. Rakosy, and C. Ramwell (2009). "Ecological impacts of early 21st century agricultural change in Europe–a review". *Journal of Environmental Management* 91, 22–46.

Storey, M.-A., N. A. Ernst, C. Williams, and E. Kalliamvakou (2020). "The Who, What, How of Software Engineering Research: A Socio-Technical Framework". *Empirical Software Engineering* 25.5, 4097–4129. DOI: 10.1007/s10664-020-09858-z.

Sundmaeker, H., C. Verdouw, S. Wolfert, and L. Pérez Freire (2016). "Internet of Food and Farm 2020". In: *Digitising the Industry: Internet of Things Connecting the Physical, Digital and Virtual Worlds*. Ed. by O. Vermesan and P. Friess. Vol. 49. River Publishers Series in Communications. Gistrup / Delft: River Publishers, 129–150. DOI: 10.1201/9781003337966-4.

Sutton, M. A., S. Reis, and K. Butterbach-Bahl (2009). "Reactive Nitrogen in Agroecosystems: Integration with Greenhouse Gas Interactions". *Agriculture, Ecosystems & Environment* 133.3–4, 135–138. DOI: 10.1016/j.agee.2009.06.008.

Suwin66 (2025). *Artificial intelligence robot harvesting strawberry in the greenhouse, 3D render*. Image licensed from Shutterstock.

Talero-Sarmiento, L. H., D. T. Parra-Sánchez, and H. L. Lamos-Diaz (2022). "Opportunities and Barriers of Smart Farming Adoption by Farmers Based on a Systematic Literature Review". In: *Proceedings of INNODOCT/22: International Conference on Innovation, Documentation and Education*. Valencia, Spain: Editorial Universitat Politècnica de València, 53–64. DOI: 10.4995/INN2022.2023.15746.

Thanos, C. (2013). "A Vision for Global Research Data Infrastructures". *Data Science Journal* 12, 71–90. DOI: 10.2481/dsj.12-043.

Top, J. L., S. Janssen, H. Boogaard, M. Knapen, and G. Şimşek-Şenel (2022). "Cultivating FAIR Principles for Agri-Food Data". *Computers and Electronics in Agriculture* 196, Article 106909. DOI: 10.1016/j.compag.2022.106909.

Tress, G., B. Tress, and G. Fry (2007). "Analysis of the barriers to integration in landscape research projects". *Land Use Policy* 24.2, 374–385. DOI: 10.1016/j.landusepol.2006.05.001.

Tuia, D., K. Schindler, B. Demir, X. X. Zhu, M. Kochupillai, S. Dzeroski, J. N. Van Rijn, H. H. Hoos, F. Del Frate, M. Datcu, V. Markl, B. Le Saux, R. Schneider, and G. Camps-Valls (2024). "Artificial Intelligence to Advance Earth Observation: A review of models, recent trends, and pathways forward". *IEEE Geoscience and Remote Sensing Magazine* 2024.4. DOI: 10.1109/MGRS.2024.3425961.

Tummers, J., A. Kassahun, and B. Tekinerdogan (2019). "Obstacles and Features of Farm Management Information Systems: A Systematic Literature Review". *Computers and Electronics in Agriculture* 157, 189–204. DOI: 10.1016/j.compag.2018.12.044.

U.S. Department of Agriculture (2012). *The Yield Forecasting Program of NASS*. U.S. Department of Agriculture. (Visited on 2022).

Vakhtang, S., H. James, S. Vaishali, P. Cheryl, A. Pramod, W. Carol J., and H. Gerrit (2019). "A multi-scale and multi-model gridded framework for forecasting crop production, risk analysis, and climate change impact studies". *Environmental Modelling & Software* 115, 144–154. DOI: https://doi.org/10.1016/j.envsoft.2019.02.006.

Van der Velde, M., C. Van Diepen, and B. Baruth (2019). "The European crop monitoring and yield forecasting system: Celebrating 25 years of JRC MARS Bulletins". *Agricultural Systems* 168, 56–57. DOI: https://doi.org/10.1016/j.agsy.2018.10.003.

Van der Werf, D. C. (2009). *Nested Systems Modeling: A Hierarchical Approach to Individual-Based Models*. Poster contribution. Wageningen University & Research.

Van Ittersum, M. K. and M. Donatelli (2003). "Modelling Cropping Systems: Highlights of the Second European Conference on Modelling Cropping Systems". *European Journal of Agronomy* 18.3–4, 187–197. DOI: 10.1016/S1161-0301(02)00104-1.

Van Ittersum, M. K., F. Ewert, T. Heckelei, J. Wery, J. Alkan Olsson, E. Andersen, I. Bezlepkina, F. Brouwer, M. Donatelli, G. Flichman, L. Olsson, A. E. Rizzoli, T. Van der Wal, J.-E. Wien, and J. Wolf (2008). "Integrated Assessment of Agricultural Systems—A Component-Based Framework for the European Union (SEAMLESS)". *Agricultural Systems* 96.1–3, 150–165. DOI: 10.1016/j.agsy.2007.07.009.

Van Kraalingen, D. W. G., M. Knapen, A. De Wit, and H. L. Boogaard (2020). "WISS a Java Continuous Simulation Framework for Agro-Ecological Modelling". In: *Environmental Software Systems. Data Science in Action*. Ed. by I. N. Athanasiadis, S. P. Frysinger, G. Schimak, and W. J. Knibbe. Cham: Springer International Publishing, 242–248.

Van Meijl, H., T. Van Rheenen, A. Tabeau, and B. Eickhout (2006). "The Impact of Different Policy Environments on Agricultural Land Use in Europe". *Agriculture, Ecosystems & Environment* 114.1, 21–38. DOI: 10.1016/j.agee.2005.11.006.

Vasilieva, N. and A. Cherepanov (2017). "Curve fitting of MODIS NDVI time series in the task of early crops identification by satellite images". *Procedia Engineering* 201, 184–195. DOI: 10.1016/j.proeng.2017.09.596.

Verburg, P. H., P. P. Schot, M. J. Dijst, and A. Veldkamp (2004). "Land Use Change Modelling: Current Practice and Research Priorities". *GeoJournal* 61.4, 309–324. DOI: 10.1007/s10708-004-4946-y.

Verweij, P. J. F. M., M. Knapen, W. De Winter, J. J. F. Wien, and J. te Roller (2010). "An IT Perspective on Integrated Environmental Modelling: The SIAT Case". *Ecological Modelling* 221.18, 2167–2176. DOI: 10.1016/j.ecolmodel.2010.01.006.

Villa, F., I. N. Athanasiadis, and A. E. Rizzoli (2009). "Modelling with Knowledge: A Review of Emerging Semantic Approaches to Environmental Modelling". *Environmental Modelling & Software* 24.5, 577–587. DOI: 10.1016/j.envsoft.2008.09.009.

Vitolo, C., Y. Elkhatib, D. Reusser, C. J. A. Macleod, and W. Buytaert (2015). "Web Technologies for Environmental Big Data". *Environmental Modelling & Software* 63, 185–198. DOI: 10.1016/j.envsoft.2014.10.007.

Von Hanstein, O. (1924). *Die Farm des Verschollenen*. Berlin: Ullstein Verlag.

– (1935). "The Hidden Colony". *Wonder Stories* 6.10. Illustrated by Frank R. Paul. Translated version of *Die Farm des Verschollenen* (1924), 1211–1227.

Wachowicz, M., L. A. Vullings, M. Van den Broek, and A. Ligtenberg (2003). *Games for interactive spatial planning: SPLASH, a prototype strategy game about water management*. Tech. rep. Alterra-rapport 667. Wageningen, The Netherlands: Alterra, Wageningen University & Research, 55.

Wang, W., A. Tolk, and W. Wang (2009). "The Levels of Conceptual Interoperability Model: Applying Systems Engineering Principles to M&S". In: *Proceedings of the 2009 Spring Simulation Multiconference (SpringSim '09)*. San Diego, CA, USA: Society for Computer Simulation International, 168:1–168:9. DOI: 10.5555/1639809.1655398.

Waters, N. M. (2017). "Tobler's First Law of Geography". In: *International Encyclopedia of Geography: People, the Earth, Environment and Technology*. Ed. by D. S. Richardson, N. Castree, M. F. Goodchild, A. Kobayashi, W. Liu, and R. A. Marston. Hoboken, NJ: Wiley-Blackwell, 1–13. DOI: 10.1002/9781118786352.wbieg1011.

Whelan, G., K. J. Castleton, J. W. Buck, B. L. Hoopes, M. A. Pelton, D. L. Strenge, G. M. Gelston, and R. N. Kickert (1997). *Concepts of a Framework for Risk Analysis in Multimedia Environmental Systems*. Tech. rep. PNNL-11748. Richland, Washington, USA: Pacific Northwest National Laboratory (PNNL), 61.

Whitcraft, A., I. Becker-Reshef, and C. Justice (2020). "NASA Harvest(ing) Earth Observations for Informed Agricultural Decisions". In: *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, 3706–3708. DOI: 10.1109/IGARSS39084.2020.9324176.

White, J. W., L. A. Hunt, K. J. Boote, J. W. Jones, J. Koo, S. Kim, C. H. Porter, P. W. Wilkens, and G. Hoogenboom (2013). "Integrated description of agricultural field experiments and production: The ICASA Version 2.0 data standards". *Computers and Electronics in Agriculture* 96, 1–12. DOI: 10.1016/j.compag.2013.04.003.

Wilkinson, M. D., M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. Van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. Van der Lei, E. Van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, and B. Mons (2016). "The FAIR Guiding Principles for scientific data management and stewardship". *Scientific Data* 3, 160018. DOI: 10.1038/sdata.2016.18.

Wolfert, S., L. Ge, C. Verdouw, and M. Bogaardt (2017). "Big data in smart farming – a review". *Agricultural systems* 153, 69–80.

World Bank (2023). *Digital Agriculture: The Future of Smart Farming*. https://www.worldbank.org/en/topic/agriculture/brief/digital-agriculture. Accessed: 2025-08-16.

Zeginis, D., E. Kalampokis, R. Palma, R. Atkinson, and K. A. Tarabanis (2024). "A Semantic Meta-Model for Data Integration and Exploitation in Precision Agriculture and Livestock Farming". *Semantic Web* 15.4, 1165–1193. DOI: 10.3233/SW-233156.

Zervas, P., N. Manouselis, P. Karampiperis, O. Hologne, S. Janssen, and J. Keizer (2018). *e-ROSA D3.7: Foresight Roadmap Paper*. Tech. rep. Horizon 2020 e-ROSA project deliverable (Grant Agreement No 730988). Agroknow, INRA, and Wageningen UR (ALTERA / DLO). DOI: 10.5281/zenodo.1479659.

Zuiderwijk, A. (2017). "Analysing Open Data in Virtual Research Environments: New Collaboration Opportunities to Improve Policy Making". *International Journal of Electronic Government Research* 13.4, 76–92. DOI: 10.4018/IJEGR.2017100105.

Acknowledgements

Completing this thesis has been a long journey–both professionally and personally–and it would not have been possible without the support, inspiration, and collaboration of many people along the way.

First and foremost, I would like to express my deep gratitude to my promotor, Prof. Dr Ioannis Athanasiadis, for his steady guidance, intellectual sharpness, and encouragement throughout this PhD journey–and for luring me into it in the first place. His vision and constructive feedback have helped shape not only this thesis, but also my broader thinking on research and artificial intelligence.

I am equally grateful to my co-promotor, Dr ir. Sander Janssen, for his practical wisdom, continuous support, and shared enthusiasm for data, models, and interoperability. His insight and experience, grounded in real-world challenges, have made this work both relevant and enjoyable. Pursuing my PhD alongside a full-time position would not have been possible without his support–and, in all fairness, he does bear some of the blame for encouraging me to go for it.

I sincerely thank all co-authors of the papers that form the basis of the core thesis chapters. Your contributions, whether in data, methods, discussions, or critical review, have been essential in bringing this work to life. Collaborating with you has been a privilege. I would also like to express my gratitude to Dr ir. Ron van Lammeren, who acted as co-promotor during the early stages of this PhD. His thoughtful advice and encouragement helped shape the initial direction of my work. Although he had to step back due to retirement, his early support remains very much appreciated. I am also deeply grateful to the scientific communities of the International Modelling and Software (iEMSs) and the Modelling and Simulation Society of Australia and New Zealand Inc. (MSSANZ), who welcomed me so kindly from the very beginning of my scientific journey. Your openness, critical engagement, and collaborative spirit have not only shaped the scientific landscape I work in but have also offered a sense of belonging that has enriched my professional path.

To all of my colleagues at Wageningen Environmental Research, thank you for creating an open, collaborative, and creative working environment. Whether we cross paths in the hallway, in code discussions, or at project meetings, your input, humour, and camaraderie

are always appreciated. I also want to acknowledge those who have since moved on to new opportunities or into retirement–your contributions continue to resonate.

I also extend my thanks to all fellow PhD candidates and postdoctoral researchers at Wageningen University, especially those in the Artificial Intelligence group and the Laboratory of Geo-information Science and Remote Sensing. Your insights, encouragement, and shared conversations, both technical and personal, have been an important source of motivation and perspective. I hope this thesis serves as a small reminder that every step forward, no matter how incremental, contributes to the collective progress we are making in science and understanding. Keep going, you are doing important work.

Finally, a very warm and heartfelt thank you to my family. To my late father, who always encouraged curiosity and critical thinking and who supported me in my early steps using programmable calculators (like the TI-59) and our very first home computer, the Acorn Atom. He helped lay the foundations of my mathematical thinking in a traditional, steady way. To my mother, who now lives with dementia, but who also passed on her gift for working with numbers and has always stood behind me with quiet strength and unwavering support. Together, they encouraged every path I chose, and their belief in me has carried me throughout this journey. And to my brother, sister, and her family: thank you for your support, even when unspoken. I dedicate this thesis to all of you–with love, gratitude, and pride.

This thesis marks the completion of a meaningful chapter in my life, coinciding with my 25th anniversary at Wageningen University & Research. I am grateful to have had the opportunity to grow, contribute, and learn within such a vibrant and mission-driven organisation.

Thank you all.

About the author

Rob Knapen was born on October 28, 1967, in the village of Elsloo (Limburg), in the southern part of the Netherlands. He is a senior AI and Research Software Engineer at Wageningen Environmental Research and a PhD candidate in the Artificial Intelligence group at Wageningen University. His doctoral research focuses on spatial data engineering, GeoAI and its applications in digital agriculture and complex environmental challenges.

Rob has a background in Computer Science and over 25 years of experience working with geospatial data, Geographic Information Systems (GIS), scalable computing, and research software engineering in environmental and agricultural science. Before joining Wageningen University & Research, he worked at the Dutch national Topographic Institute (Topografische Dienst Nederland) in Emmen–now part of the Cadastre–where he contributed to national and international digital geospatial infrastructure efforts.

Throughout his career, Rob has contributed to numerous large European research projects. Notable achievements include his involvement in the development of the OGC Open Modeling Interface (OpenMI), the Dutch national AgroDataCube, and the Open Agriculture KPI Calculator. His work consistently bridges technical innovation with real-world impact, particularly in support of sustainable food systems and responsible environmental management.

He is also a member of the International Environmental Modelling & Software Society (iEMSs), reflecting his ongoing engagement with the broader research community in modelling, interoperability, and environmental decision support.

Fittingly, the completion and defence of this thesis coincides with Rob's 25th anniversary at Wageningen University & Research–a milestone that reflects both his long-standing commitment and evolving role at the intersection of science, technology, and society.

Outside of work, Rob enjoys scuba diving and photography.

Peer-reviewed Journal Publications

Knapen, M. J. R., A. De Wit, E. Buyukkaya, P. Petrou, D. Paudel, S. Janssen, and I. Athanasiadis (2025). "Efficient and scalable crop growth simulations using standard big data and distributed computing technologies". *Computers and Electronics in Agriculture* 236, 110392. DOI: 10.1016/j.compag.2025.110392.

Pook, T., J. Vandenplas, J. C. Boschero, E. Aguilera, K. Leijnse, A. Chauhan, Y. Bouzembrak, M. J. R. Knapen, and M. Aldridge (2025). "Assessing the potential of quantum computing in agriculture". *Computers and Electronics in Agriculture* 235, 110332. DOI: 10.1016/j.compag.2025.110332.

Knapen, M. J. R., R. M. Lokers, and S. J. C. Janssen (2023). "Evaluating the D4Science virtual research environment platform for agro-climatic research". *Agricultural Systems* 210, 103706. DOI: 10.1016/j.agsy.2023.103706.

Top, J., S. Janssen, H. Boogaard, M. J. R. Knapen, and G. Şimşek-Şenel (2022). "Cultivating FAIR principles for agri-food data". *Computers and Electronics in Agriculture* 196. Open access, CC BY, 106909. DOI: 10.1016/j.compag.2022.106909.

Assante, M., A. Boizet, L. Candela, D. Castelli, R. Cirillo, G. Coro, E. Fernández, M. Filter, L. Frosini, T. Georgiev, G. Kakaletris, P. Katsivelis, M. J. R. Knapen, L. Lelii, R. M. Lokers, F. Mangiacrapa, N. Manouselis, P. Pagano, G. Panichi, L. Penev, and F. Sinibaldi (2021). "Realizing virtual research environments for the agri-food community: The AGINFRA PLUS experience". *Concurrency and Computation: Practice and Experience* 33.19. First published online: 25 November 2020, e6087. DOI: 10.1002/cpe.6087.

De Wit, A., H. Boogaard, D. Fumagalli, S. Janssen, M. J. R. Knapen, D. Van Kraalingen, I. Supit, R. Van der Wijngaart, and K. Van Diepen (2019). "25 years of the WOFOST cropping systems model". *Agricultural Systems* 168. First online: 20 Jul 2018; Open access (CC BY), 154–167. DOI: 10.1016/j.agsy.2018.06.018.

Lokers, R., M. J. R. Knapen, S. Janssen, Y. Van Randen, and J. Jansen (2016). "Analysis of Big Data technologies for use in agro-environmental science". *Environmental Modelling & Software* 84, 494–504. DOI: 10.1016/j.envsoft.2016.07.017.

Porter, C. H., C. Villalobos, D. Holzworth, R. Nelson, J. W. White, I. N. Athanasiadis, S. Janssen, D. Ripoche, J. Cufi, D. Raes, M. Zhang, M. J. R. Knapen, R. Sahajpal, K. Boote, and J. W. Jones (2014). "Harmonization and translation of crop modeling data to ensure interoperability". *Environmental Modelling & Software* 62. Available online: 30 Oct 2014, 495–508. DOI: 10.1016/j.envsoft.2014.09.004.

Knapen, M., S. Janssen, O. Roosenschoon, P. Verweij, W. De Winter, M. Uiterwijk, and J.-E. Wien (2013). "Evaluating OpenMI as a model integration platform across disciplines". *Environmental Modelling & Software* 39. Online first: 2012, 274–282. DOI: 10.1016/j.envsoft.2012.06.011.

Janssen, S., I. N. Athanasiadis, I. Bezlepkina, M. Knapen, H. Li, I. Pérez Domínguez, A. E. Rizzoli, and M. K. Van Ittersum (2011). "Linking models for assessing agricultural land use change". *Computers and Electronics in Agriculture* 76.2, 148–160. DOI: 10.1016/j.compag.2010.10.011.

Verweij, P. J. F. M., M. Knapen, W. P. De Winter, J.-E. Wien, J. A. Te Roller, S. Sieber, and J. M. L. Jansen (2010). "An IT perspective on integrated environmental modelling: The SIAT case". *Ecological Modelling* 221.18, 2167–2176. DOI: 10.1016/j.ecolmodel.2010.01.006.

Alkan Olsson, J., C. Bockstaller, L. M. Stapleton, F. Ewert, M. Knapen, O. Thérond, G. Geniaux, S. Bellon, T. Pinto-Correia, N. Turpin, and I. Bezlepkina (2009). "A goal oriented indicator framework to support integrated assessment of new policies for agri-environmental systems". *Environmental Science & Policy* 12.5, 562–572. DOI: 10.1016/j.envsci.2009.01.012.

Ewert, F., M. K. Van Ittersum, I. Bezlepkina, O. Therond, E. Andersen, H. Belhouchette, C. Bockstaller, F. Brouwer, T. Heckelei, S. Janssen, M. J. R. Knapen, M. Kuiper, K. Louhichi, J. Alkan Olsson, N. Turpin, J. Wery, J. E. Wien, and J. Wolf (2009). "A methodology for enhanced flexibility of integrated assessment in agriculture". *Environmental Science & Policy* 12.5, 546–561. DOI: 10.1016/j.envsci.2009.02.005.

Janssen, S., F. Ewert, H. Li, I. N. Athanasiadis, J. J. F. Wien, O. Thérond, M. Knapen, I. Bezlepkina, J. Alkan-Olsson, A. E. Rizzoli, H. Belhouchette, M. Svensson, and M. K. van Ittersum (2009). "Defining assessment projects and scenarios for policy support: Use of ontology in Integrated Assessment and Modelling". *Environmental Modelling & Software* 24.12, 1491–1500. DOI: 10.1016/j.envsoft.2009.04.009.

Other Scientific Publications

Knapen, M. J. R., R. M. Lokers, L. Candela, and S. J. C. Janssen (2020). "AGINFRA PLUS: Running Crop Simulations on the D4Science Distributed e-Infrastructure". In: *Environmental Software Systems. Data Science in Action*. Ed. by I. N. Athanasiadis, S. P. Frysinger, G. Schimak, and W. J. Knibbe. Vol. 554. IFIP Advances in Information and Communication Technology. Springer, Cham, 81–89. DOI: 10.1007/978-3-030-39815-6_8.

Lokers, R. M., M. J. R. Knapen, L. Candela, S. Hoek, and W. M. L. Meijninger (2020). "Using Virtual Research Environments in Agro-Environmental Research". In: *Environmental Software Systems. Data Science in Action*. Ed. by I. N. Athanasiadis, S. P. Frysinger, G. Schimak, and W. J. Knibbe. Vol. 554. IFIP Advances in Information and Communication Technology. Springer, Cham, 115–121. DOI: 10.1007/978-3-030-39815-6_11.

Van Kraalingen, D. W. G., M. J. R. Knapen, A. De Wit, and H. L. Boogaard (2020). "WISS a Java Continuous Simulation Framework for Agro-Ecological Modelling". In:

Environmental Software Systems. Data Science in Action. Ed. by I. N. Athanasiadis, S. P. Frysinger, G. Schimak, and W. J. Knibbe. Vol. 554. IFIP Advances in Information and Communication Technology. Springer, Cham, 242–248. DOI: 10.1007/978-3-030-39815-6_23.

Janssen, H., S. J. C. Janssen, M. J. R. Knapen, W. M. L. Meijninger, Y. Van Randen, I. J. la Riviere, and G. J. Roerink (2018). *AgroDataCube: A Big Open Data collection for Agri-Food Applications*. Dataset. Wageningen Environmental Research. DOI: 10.18174/455759.

Knapen, M. J. R., R. M. Lokers, Y. van Randen, S. J. C. Janssen, and H. Janssen (2018). "AgroDataCube and AGINFRA+: Operationalising Big Data for Agricultural Informatics". In: *Scientific Symposium FAIR Data Sciences for Green Life Sciences (FAIRdata 2018)*. Conference abstract. Wageningen University & Research. Wageningen, Netherlands, 1. DOI: 10.18174/FAIRdata2018.16273.

Lokers, R., Y. Van Randen, M. Knapen, S. Gaubitzer, S. Zudin, and S. Janssen (2015). "Improving Access to Big Data in Agriculture and Forestry Using Semantic Technologies". In: *Metadata and Semantics Research*. Ed. by E. Garoufallou, R. J. Hartley, and P. Gaitanou. Vol. 544. Communications in Computer and Information Science. Cham: Springer, 369–380. DOI: 10.1007/978-3-319-24129-6_32.

Porter, C. H., C. Villalobos, D. Holzworth, R. Nelson, J. W. White, I. N. Athanasiadis, M. Zhang, S. J. C. Janssen, M. J. R. Knapen, J. W. Jones, K. J. Boote, J. Hargreaves, and J. M. Antle (2015). "Data Interoperability Tools for Regional Integrated Assessments". In: *Handbook of Climate Change and Agroecosystems: The Agricultural Model Intercomparison and Improvement Project (AgMIP) Integrated Crop and Economic Assessments, Part 2*. Ed. by C. Rosenzweig and D. Hillel. Vol. 3. ICP Series on Climate Change Impacts, Adaptation, and Mitigation. London: Imperial College Press, 147–171. DOI: 10.1142/9781783265640_0006.

Knapen, M. J. R., T. Hüsing, K. Jacob, Y. Van Randen, S. Reis, O. R. Roosenschoon, and S. J. C. Janssen (2014). "Metadata extraction using semantic and Natural Language Processing techniques". In: *Proceedings of the 7th International Congress on Environmental Modelling and Software (iEMSs 2014): Bold Visions for Environmental Modeling*. Ed. by D. P. Ames, N. W. T. Quinn, and A. E. Rizzoli. Conference abstract; no article-level DOI. San Diego, California, USA: International Environmental Modelling and Software Society (iEMSs), 385–391.

Lokers, R. M., S. Konstantopoulos, A. Stellato, M. J. R. Knapen, and S. J. C. Janssen (2014). "Designing innovative Linked Open Data and semantic technologies for agro-environmental modelling". In: *Proceedings of the 7th International Congress on Environmental Modelling and Software (iEMSs 2014): Bold Visions for Environmental Modeling*. Ed. by D. P. Ames, N. W. T. Quinn, and A. E. Rizzoli. Conference abstract;

no article-level DOI. San Diego, California, USA: International Environmental Modelling and Software Society (iEMSs).

Verweij, P. J. F. M., M. Winograd, M. Pérez-Soba, M. J. R. Knapen, and Y. van Randen (2012). "QUICKScan: A pragmatic approach to decision support". In: *Managing Resources of a Limited Planet: Pathways and Visions under Uncertainty — Proceedings of the 6th Biennial Meeting of the International Environmental Modelling and Software Society (iEMSs 2012)*. Ed. by R. Seppelt, A. A. Voinov, S. Lange, and D. Bankamp. No article-level DOI. Leipzig, Germany: International Environmental Modelling and Software Society (iEMSs), 1877–1884.

Knapen, M. J. R., S. J. C. Janssen, O. R. Roosenschoon, and P. J. F. M. Verweij (2011). "Evaluating and improving OpenMI as a model integration platform across disciplines". In: *MODSIM2011, 19th International Congress on Modelling and Simulation: Sustaining Our Future—Understanding and Living with Uncertainty*. Ed. by F. Chan, D. Marinova, and R. S. Anderssen. No article-level DOI assigned. Perth, Australia: Modelling, Simulation Society of Australia, and New Zealand (MSSANZ), 1223–1229.

Bezlepkina, I., M. Adenäuer, M. H. Kuiper, S. J. C. Janssen, M. J. R. Knapen, A. Kanellopoulos, F. M. Brouwer, J. J. F. Wien, J. Wolf, and M. K. van Ittersum (2010). "Using the SEAMLESS Integrated Framework for ex-ante assessment of trade policies". In: *Towards effective food chains: Models and applications*. Ed. by J. Trienekens, J. Top, J. van der Vorst, and A. Beulens. Wageningen: Wageningen Academic Publishers, 251–271. DOI: 10.3920/9789086867059_015.

Donatelli, M., G. Russell, A. E. Rizzoli, M. Acutis, M. Adam, I. N. Athanasiadis, M. Balderacchi, L. Bechini, H. Belhouchette, G. Bellocchi, J.-E. Bergez, M. Botta, E. Braudeau, S. Bregaglio, L. Carlini, E. Casellas, F. Celette, E. Ceotto, M. H. Charron-Moirez, R. Confalonieri, M. Corbeels, L. Criscuolo, P. Cruz, A. di Guardo, D. Ditto, C. Dupraz, M. Duru, D. Fiorani, A. Gentile, F. Ewert, C. Gary, E. Habyarimana, C. Jouany, K. Kansou, M. Knapen, G. Lanza Filippi, P. A. Leffelaar, L. Manici, G. Martin, P. Martin, E. Meuter, N. Mugueta, R. Mulia, M. van Noordwijk, R. Oomen, A. Rosenmund, V. Rossi, F. Salinari, A. Serrano, A. Sorce, G. Vincent, J.-P. Theau, O. Thérond, M. Trevisan, P. Trevisiol, F. K. Van Evert, D. Wallach, J. Wery, and A. Zerourou (2010). "A Component-Based Framework for Simulating Agricultural Production and Externalities". In: *Environmental and Agricultural Modelling: Integrated Approaches for Policy Impact Assessment*. Ed. by F. M. Brouwer and M. K. Van Ittersum. Dordrecht: Springer, 63–108. DOI: 10.1007/978-90-481-3619-3_4.

Donchyts, G., S. Hummel, S. Vaneček, J. Grooss, A. Harper, M. Knapen, J. Gregersen, P. Schade, A. Antonello, and P. Gijsbers (2010). "OpenMI 2.0 – What's new?" In: *Modelling for Environment's Sake: Proceedings of the 5th International Congress on Environmental Modelling and Software (iEMSs 2010)*. Ed. by D. A. Swayne, W. Yang, A. A. Voinov, A. E. Rizzoli, and T. Filatova. Vol. 2. Conference paper; no article-level

DOI. Ottawa, Ontario, Canada: International Environmental Modelling and Software Society (iEMSs).

Gijsbers, P., S. Hummel, S. Vaneček, J. Grooss, A. Harper, M. Knapen, J. Gregersen, P. Schade, A. Antonello, and G. Donchyts (2010). "From OpenMI 1.4 to 2.0". In: *Modelling for Environment's Sake: Proceedings of the 5th International Congress on Environmental Modelling and Software (iEMSs 2010)*. Ed. by D. A. Swayne, W. Yang, A. A. Voinov, A. E. Rizzoli, and T. Filatova. Vol. 2. Conference paper; no article-level DOI. Ottawa, Ontario, Canada: International Environmental Modelling and Software Society (iEMSs), 1081–1088.

Knapen, M. J. R., P. J. F. M. Verweij, and S. J. C. Janssen (2010). "Agilists and the Art of Integrated Assessment Tool Development". In: *Modelling for Environment's Sake: Proceedings of the 5th Biennial Conference of the International Environmental Modelling and Software Society (iEMSs 2010)*. Ed. by D. A. Swayne, W. Yang, A. A. Voinov, A. E. Rizzoli, and T. Filatova. Vol. 3. No article-level DOI assigned. Ottawa, Canada: International Environmental Modelling and Software Society (iEMSs).

Wien, J.-E., A. E. Rizzoli, M. Knapen, I. N. Athanasiadis, S. Janssen, L. Ruinelli, F. Villa, M. Svensson, P. Wallman, B. Jonsson, and M. K. Van Ittersum (2010). "A Web-Based Software System for Model Integration in Impact Assessments of Agricultural and Environmental Policies". In: *Environmental and Agricultural Modelling: Integrated Approaches for Policy Impact Assessment*. Ed. by F. M. Brouwer and M. K. Van Ittersum. Dordrecht: Springer, 207–234. DOI: 10.1007/978-90-481-3619-3_9.

Knapen, M. J. R., P. J. F. M. Verweij, J. J. F. Wien, and S. Hummel (2009). "OpenMI – The universal glue for integrated modelling?" In: *18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation: Interfacing Modelling and Simulation with Mathematical and Computational Sciences*. Ed. by R. S. Anderssen, R. D. Braddock, and L. T. H. Newham. No article-level DOI assigned. Cairns, Australia: Modelling et al., 895–901.

Janssen, S. J. C., J. J. F. Wien, H. Li, I. N. Athanasiadis, F. Ewert, M. J. R. Knapen, D. Huber, O. Thérond, A. E. Rizzoli, H. Belhouchette, M. Svensson, and M. K. van Ittersum (2007). "Defining projects and scenarios for integrated assessment modelling using ontology". In: *MODSIM 2007 International Congress on Modelling and Simulation: Land, Water and Environmental Management—Integrated Systems for Sustainability*. Ed. by L. Oxley and D. Kulasiri. Refereed conference paper; no article-level DOI. Christchurch, New Zealand: Modelling, Simulation Society of Australia, and New Zealand (MSSANZ), 2055–2061.

Knapen, M. J. R., P. J. F. M. Verweij, and J. J. F. Wien (2007). "Applying Enterprise Application Architectures in Integrated Modelling". In: *MODSIM 2007 International Congress on Modelling and Simulation*. Ed. by L. Oxley and D. Kulasiri. No article-level

DOI. Christchurch, New Zealand: Modelling, Simulation Society of Australia, and New Zealand, 798–804.

Verweij, P. J. F. M., M. J. R. Knapen, and J. J. F. Wien (2007). "The Use of OpenMI in Model Based Integrated Assessments". In: *MODSIM 2007 International Congress on Modelling and Simulation*. Ed. by L. Oxley and D. Kulasiri. No article-level DOI assigned. Christchurch, New Zealand: Modelling, Simulation Society of Australia, and New Zealand, 1166–1171.

Wien, J. J. F., M. J. R. Knapen, S. J. C. Janssen, P. J. F. M. Verweij, I. N. Athanasiadis, H. Li, A. E. Rizzoli, and F. Villa (2007). "Using ontology to harmonize knowledge concepts in data and models". In: *MODSIM 2007 International Congress on Modelling and Simulation: Land, Water and Environmental Management—Integrated Systems for Sustainability*. Ed. by L. Oxley and D. Kulasiri. No article-level DOI assigned. Christchurch, New Zealand: Modelling, Simulation Society of Australia, and New Zealand, 2584–2590.

Van der Wal, T., M. J. R. Knapen, M. Svensson, I. N. Athanasiadis, and A. E. Rizzoli (2005). "Trade-offs in the design of cross-disciplinary software systems". In: *Proceedings of the 16th International Congress on Modelling and Simulation (MODSIM05): Advances and Applications for Management and Decision Making*. Ed. by A. Zerger and R. M. Argent. No article-level DOI assigned. Melbourne, Australia: Modelling, Simulation Society of Australia, and New Zealand, 732–737.

PhD candidate's name:	Rob Knapen
First promotor:	Prof. Dr I. Athanasiadis
Title of PhD thesis:	Spatial Data Engineering for Digital Agriculture
Date of public defence:	10 October 2025

Chapter 1 General introduction. The overarching research theme and the societal context were initially outlined in discussions with my promotor. In addition, I defined the research problem and formulated the core research objectives. I authored the full text and figures and revised the chapter based on feedback from my promotor and co-promotor.

Chapter 2 Connecting data. This chapter is based on a publication with shared first authorship. I co-developed the research question, contributed to the assessment of big data technologies in agro-environmental science, and contextualised the findings within spatial data engineering for agriculture. My contributions included developing the methodology, performing data analysis, writing key parts of the manuscript, and coordinating revisions.

Chapter 3 Connecting models. I was the main author of this chapter. I defined the research objectives, evaluated the OpenMI standard as a model integration framework, and led the analysis of its applicability across disciplines. I wrote the manuscript and developed supporting figures, with co-authors providing reviews and additional insights. I also incorporated all the revisions and addressed the comments of the reviewers.

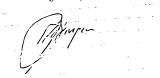
Chapter 4 Connecting systems. I led the research and writing for this chapter, designed the experimental setup, implemented the prototype system, and led and performed the analysis. I drafted the manuscript and incorporated the contributions from all co-authors. I addressed the reviewers' comments and wrote the revised version of the manuscript.

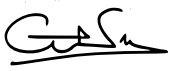
Chapter 5 Connecting researchers. I supported the configuration and custom development of VRE components in collaboration with D4Science researchers and developed the crop growth simulation functionality integrated into the VRE. I contributed to the manuscript by writing and reviewing the parts that describe the technical implementation and integration work. I led the revision of the manuscript and addressed all the comments of the reviewers.

Chapter 6 General discussion. I conceptualised and authored this integrative synthesis independently. The text was revised once following comments from my promotor and co-promotor.

Use of Generative AI. I used Overleaf's integrated Writefull tools to enhance grammar, phrasing, and scientific writing in the Introduction, Synthesis, and Summary, using its AI language model. No content was autonomously generated, and all AI-suggested revisions were subjected to a critical review and manual editing. I assume full responsibility for the final content and interpretations in this thesis.

Date: May 5, 2025

Signature PhD candidate: 

Signature promotor for agreement: 

PE&RC Training and Education Statement

With the training and education activities listed below the PhD candidate has complied with the requirements set by the Graduate School for Production Ecology and Resource Conservation (PE&RC) which comprises of a minimum total of 30 ECTS (= 20 weeks of activities)

Review/project proposal (6 ECTS)
- Spatial Data Engineering for Digital Agriculture

Post-graduate courses (14.15 ECTS)
- Minds and Machines, MITx (2021)
- Fundamentals Deep Learning, VBTI (2021)
- Data Engineering online courses, Rockthejvm.com (2020-2022)

Deficiency, refresh, brush-up courses (1.75 ECTS)
- Effective Programming in Scala, EPFL / Coursera (2021)

Invited review of journal manuscripts (10 ECTS)
- Environmental Modelling & Software, Geospatial data processing, Machine Learning (2022)
- Computers and Electronics in Agriculture, Big Data analytics for forestry (2018)
- Environmental Modelling & Software, Serious-gaming, decision support systems (2018)
- Environmental Modelling & Software, Component based modelling frameworks (2017)
- Environmental Modelling & Software, Model integration, spatial data analytics (4 papers reviewed) (2016)
- Environmental Modelling & Software: Top 10 Reviewer Honourable Mention Semantic technology, integrated assessment modelling, Big Data processing and analytics (2011-2016)
- Environmental Modelling & Software: Outstanding Reviewer Award 2011, Agile software development, model integration frameworks, OpenMI, data analytics (2004-2011)

Competence, skills and career-oriented activities (1.8 ECTS)
- "Projectmatig Creëeren" course, WUR (<2010)
- Participation in internal team building and team strategy workshops on Earth Informatics and Remote Sensing, WUR (2001–2025)
- Training project acquisition skills, WUR (2005)
- SCRUM Master training, WUR (2017)

Scientific Integrity/Ethics in science activities (0.3 ECTS)
- Standard scientific integrity and ethics session, WUR (2020)

PE&RC Annual meetings, seminars and PE&RC weekend/retreat (0.3 ECTS)
- PE&RC Day – Artificial Intelligence and Sustainability (2022)

National scientific meetings, local seminars, and discussion groups (7.7 ECTS)
- NMDC "Carrousel Big Data" (2018)
- Symposium "FAIR Data Science for Green Life Sciences" (2018)
- EU research project meetings – AgINFRA+ (2017-2019)
- EU research project meetings – Cybele (2019-2022)

International symposia, workshops and conferences (5.4 ECTS)
- ISESS 2020 presentation, Wageningen (2020)
- iEMSs 2020 presentation, Online (2020)
- iEMSs 2022 presentation, Brussels (2022)

Societally relevant exposure (4.9 ECTS)
- Article about AgroDataCube for Geo-Info magazine (2018)
- Participated in and helped organise various digital agriculture and remote sensing (ESA) related hackathons (2014, 2015, 2018)

Committee work (10.5 ECTS)
- Organised and chaired iEMSs conference session (2020)
- OpenMI Association / Technical Committee (2010–2018)
- Chair of "ondernemingsraad" for Wageningen Software Labs (<2010)

Lecturing/Supervision of practicals/tutorials (1.5 ECTS)
- Mobile apps lecture (2012-2017)

Part of the research described in this thesis was financially supported by several grants from the European Union. For details, please refer to the acknowledgements in the individual chapters.

The financial support provided by Wageningen University for the printing of this thesis is gratefully acknowledged.

Cover design by Rob Knapen, illustration © Innoria / Shutterstock.com

Printed by ProefschriftMaken.nl