Aligning Software Engineering and Artificial Intelligence with Transdisciplinary

Wangai Mambo

United States International University Africa Nairobi, Kenya, Email:mambon20@aol.com

Received 21 October, 2021; Revised 3 January, 2022; Accepted 4 January, 2022
Available online 4 January, 2022 at www.atlas-tjes.org, doi: 10.22545/2022/00177

This study examined AI and SE transdisciplinarity to find ways of aligning them to enable the development of AI-SE transdisciplinary theory. A literature review and analysis method were used. The findings are AI and SE transdisciplinarity is tacit with islands within and between them that can be linked to accelerating their transdisciplinary orientation by codification, internally developing, and externally borrowing and adapting transdisciplinary theories. Lack of theory has been identified as the major barrier to maturing the two disciplines as engineering disciplines. Creating AI and SE transdisciplinary theory would contribute to maturing AI and SE engineering disciplines. Implications of the study are transdisciplinary theory can support mode 2 and 3 AI and SE innovations; provide an alternative for maturing two disciplines as engineering disciplines. Study's originality it's first in SE, AI, or their intersections.

Keywords: Field, intelligent software engineering, transdisciplinary artifacts, trans-field.

1 Introduction

Discipline and field are often used interchangeably [1], which is also the case in computing disciplines. Discipline has been defined as an area of academic study which constitutes teaching and research in academic community as a whole [Oxford education dictionary 2015] a group of people in particular field whose work is directed toward study of phenomena and development of theories [Oxford dictionary of occupational science and occupational therapy 2017] and a branch of knowledge, learning or teaching [Open education sociology dictionary). Disciplines have corresponding domains of research, practice and communities of researchers and practitioners. This study uses term discipline as unit of knowledge that cannot be divided into parts (fields) that have all discipline core concepts.

Disciplines sometimes start as single fields but with growth branch into two or more fields. There is a broader increased diversification and specialization of knowledge that blurs traditional boundaries and challenges organization of science [2]. Discipline legitimatization depends on disciplinary politics. Politics can be of two types paradigmatic thinking that ring fences disciplines of research philosophies under their epistemologies and methodologies; and academic seclusions [3]. Transdisciplinarity overcomes these weaknesses. Discipline formation is based on consensus on what knowledge is shared, shared knowledge accumulates, legitimatization through institutions in latter formation stages and formation of dense network of researchers [2]. The same stages apply to field formation. Discipline's networks are made of communities of researchers and practitioners.

Transdisciplinarity is based on principles of searching for unity of knowledge, transcending and integrating paradigms, participatory development and focusing on relevant social issues [4]. These principles are important for SE and AI as often software development spans multiple disciplines and some like mobile phone software are developed to be used by everyone. To become transdisciplinary an individual requires: institutional transgression, respect for cultural diversity, courage to at times put ones discipline aside and depend on other disciplines [5]. These enable one to access new knowledge, perspectives and experiences. Experimentation and using a cycle that builds confidence of adopters of SE methods is one way to increase method innovations adoption [6]. Making software technology innovation have advantages over software it substitutes, making it compatible with adopter believes and values, reducing its complexity, allowing limited experimentation by adopter before adopting [7] build confidence of adopters to adopt. Transdisciplinarity is relatively new in computing, consequently should be diffused in ways build confidence of adopting professionals.

Relationship between transdisciplinarity and SE is just beginning to be explored [8]. The exploration is important because transdisciplinary characteristics of SE discipline and artifacts are tacit making it difficult to leverage and develop theories of transdisciplinarity. Second SE has been exploring how to develop SE theories to become a mature engineering discipline. Third disciplinary specialization has dominated academic knowledge development resulting in less development of multiple disciplines systemic aspects of knowledge. Artificial intelligence at broad level on other hand is an attempt to mimic the brain making it transdisciplinary [8]. Developing software that requires knowledge systematic multiple disciplines solutions thus is developed through learning from doing that is not as efficient and effective and knowledge driven problem solving. The main argument for transdisciplinarity is that reductionism used to produce disciplinary knowledge is good for problems requiring disciplinary knowledge but not problem requiring systemic transdisciplinary solutions. All disciplines solving systemic problems require solutions that unify knowledge. Transdisciplinary is not an antithesis of disciplinarily but a synthesis of transdisciplinarity and systemism. Its leverages both disciplinary and transdisciplinary systemic knowledge [5, 8] based on quantum duality [9]. Software engineering and artificial intelligence disciplines will be referred to as software disciplines (SWD). The terms software, software discipline and software development from here be used to refer to AI and SE, otherwise if referring to one of the disciplines they will be clarified.

Strategy aligns different engineering, business and computing research and practice activities to create synergy. Organizations align their efforts and activities by their strategy and vision. The alignment drives efforts and project in direction of vision making them complementary and cumulative. In same way aligning SWD AI and SE transdisciplinary efforts and projects with transdiscipline will make them complementary and cumulate in direction of achieving transdisciplinary vision. They will contribute knowledge to transdisciplinary discipline as well as benefit from transdiscipline knowledge.

Developing software involves creating a software engineering (SE) or artificial intelligence (AI) domain solution often for a different discipline problem domain. SE and AI are software disciplines (SWD). Some problems from two disciplines are solved with transdisciplinary approaches, and others by disciplinary approaches. Published literature shows SE and AI needs theories to become a mature engineering disciplines'. To develop AI and SE system level theories transdisciplinary tacit knowledge needs to be codified, systematized and relationships between transdisciplinary transdiscipline and software disciplines or established.

Study objectives are to: identify tacit transdisciplinary AI and SE fields, methods, elements; discover how they can be aligned with transdisciplinary and assess need for creating a transdisciplinary AI-SE trans-field specializing in transdisciplinarity research and theory development.

The rest of paper is structured as follows: opportunities for interactions between transdisciplinarity AI, SE and society section, infusing AI and SE transdisciplinary lenses section and towards a transdisciplinary AI-SE trans-field section.

2 Opportunities for Interactions between Transdisciplinarity AI, SE and Society

Society needs transdisciplinary software engineers who are able to implement useful elements from other disciplines in their products and services [10]. Software engineers are being trained to be artificial intelligence engineers [11]. These AI-SE engineers are transdisciplinary interpreters for field of intelligent SE are prepared to develop knowledge based software with SE and AI components. Intelligent SE field aims to make AI and SE complementary [12, 13, 14] and has two sub-fields: SE for AI (SE4E) and AI for SE (AI4SE). Demand Intelligent SE is increasing and is likely to increase exponentially in the near future. Intelligent System Engineering aims to make system engineering (SYE) and AI complementary with two emerging areas: SYE for AI (SYE4AI) and AI for SYE (AI4SYE) [15]. System engineering and SE are overlapping disciplines and to intelligent areas will benefit from research and lessons learned from two disciplines. Human domain knowledge can be used to develop AI techniques [12] and AI inspired design as inspiration for developing human computation techniques. Human computations are intelligent systems for organizing humans to carry out computation [16]. Any action, activity or phenomena that has inputs/causes and outputs/effects is a kind of computation.

Software has become pervasive and affects all aspects of our lives. A transdisciplinary software trans-field would unify knowledge from AI, SE fields, develop some transdisciplinary knowledge within intersection of two disciplines and across disciplines, contribute knowledge to transdisciplinarity and enable unification of knowledge for software development from different groups in society.

Software reuse has been one of promising areas to drastically reduce cost of software, however this has largely focused on code and design patterns, but transdisciplinary can enable extending this to planning and design software development activities and to disciplines. Bioinspired software design reuses biological solutions like organism genetics in SE. Transdisciplinarity is based on idea of developing a solution once and ideally using solution in all disciplines. Creating a software transdisciplinary trans-field specializing in transdisciplinary research, theory and artifact development would make software disciplines better contributors and beneficiaries of internally and externally developed transdisciplinary solutions. This can ride on knowledge of movements trying to integrate transdisciplinarity in bioinspired design [17] and mechatronics [18].

Software designers are few and most software projects don't have designers. Design role has become role of all software development professionals. In agile methods software development even customers learn some design as they are active participants in all steps of project. Having a transdisciplinary SE community would help developing knowledge, skills and reproduce and enhance them through interaction [19]. Everyone designs. Mass innovation aims to provide design techniques to masses for innovation [20] and grassroots innovation are innovations movements by ordinary people using everyday knowledge are driven by transdisciplinarity. Transdisciplinary helps these two types of movements to move across, beyond and between different types of boundaries, people and to unify different types of knowledge.

Mobile phones have become widely used making most society members mobile technology users. Technology users have incredible capabilities and under the right circumstances want to do and act like designers [21] transdisciplinarity provides good environment to achieve this. Software designers transport and combine knowledge from multiple domains utilizing multiple skills they possess [22]. These knowledge and skills is useful for transdisciplinary if they can be systemized and represented in transdisciplinary language.

Transdisciplinary research has become integral part of innovation and solving problems in private, public sectors and personal life by helping overcome mismatch between knowledge production in academia and knowledge required to solve societal problems [23]. Transdisciplinarity acknowledges disciplines (specialization) and provides means to overcome its shortcomings to meet societal needs [8]. Leveraging transdisciplinary knowledge by Society increasingly depends on software.

Albert Einstein famous saying scientific way of forming concepts is not very different from every day of forming concepts is same as notions and patterns of design that are fundamental and cut across all disciplines and provide insight and logic necessary to apply knowledge to any problem [24]. Ibid. assertion has two implications for transdisciplinarity. The first is that design is transdisciplinary and the second is that everyday concept formation used ordinary people and experts in areas outside their expertise can be improved by transdisciplinarity. However differences identified in Einstein's saying of their lacking systemic choice of experimental methods, precision in concepts and conclusions must be remedied in transdisciplinary process.

Most software engineers are likely to be receptive to transdisciplinarity because they frequently get inspiration from other disciplines. Creating transdisciplinary trans-field would institutionalize and create a community focusing on transdisciplinarity in computing.

3 Infusing Software Engineering and Artificial Intelligence with Transdisciplinarity Lenses

Aligning SE, AI and transdisciplinarity will make two disciplines and transdisciplinary knowledge more complementary. Transdisciplinary methodology and theory aligned with AI-SE enables accumulation research and knowledge by making external borrowing and adapting as well us internal AI-SE transdisciplinary development.

3.1 Nature of Software Engineering and Artificial Intelligence

SE is a discipline [25], divided into fields each dealing with subset of SE phenomena. It relies on concepts from computer science, mathematics, natural science, engineering, social science, cognitive science, economics, library science, general management, management science, political science [26] and artificial intelligence. AI is a science and engineering discipline [27] divided into fields' robotics, natural language processing, machine learning, computer vision, automated reasoning, and knowledge representation [28]. In recent decades view of AI as science that studies intelligence has been partially superseded by AI as engineering discipline unifying with other computing discipline to integrate AI techniques [29]. AI based on concepts from economics, general psychology, cognitive psychology, computer engineering, control theory and cybernetics, linguistics, mathematics, neural science [28], expert systems and software engineering. AI is a computing discipline although it is not included Institute of electronic, electronic engineers/ Association of computing machinery (IEEE/ACM) computing disciplines curriculum. Transdisciplinarity is not only useful in solving problems in different disciplines but also problems in domain and field silos within a discipline [8]. Creating unified classification system within computing disciplines helps them share knowledge [30] and unifying disciplines terminologies is important step to creating a transdisciplinary language.

Fields have been defined as divisions that correspond to a discipline [1]. Fields are parts of discipline that have a subset of discipline's core concepts. In addition to subset discipline's core concepts, fields have their unique core and none core concepts. Concepts are integrated into theories. SE is a system with transdisciplinary and disciplinary subsystems (fields), while all AI fields are transdisciplinary. Examples of transdisciplinary unified AI and SE fields are search based SE [31], intelligent SE [13] and design patterns. SE has transdisciplinary field fragments with potential to develop into fields like TRIZ for software consisting of a contradiction matrix [32] and inventive principles [33] and bioinspired SE. SE and AI have community wide agreed knowledge, research methodologies, while fields have respective sub communities. Theories are aggregated into fields that are accumulated into disciplines [1] and disciplines are aggregated into academic body of knowledge.

AI psychology, design and philosophy transdisciplinarity is mind design [34]:

> Mind design understands mind that is intellect and thinking needed by design to build what works oriented towards structure and mechanism. AI Mind design attempts to construct intelligent artifacts, systems with minds of their own. Mind design is psychology by reverse engineering.

AI transdisciplinary unifies all branches of knowledge [27]:

> AI science can be described as "synthetic psychology", "experimental philosophy", or "computational epistemology". History of technology indicates humans have used technologies to model themselves and several technologies such as computers and switching systems have been proposed as technology metaphors of intelligence and mechanisms for modeling mind.

AI is transdisciplinary and has arguably had more profound impact on philosophy discourse than any other discipline [29]. The building of intelligent robots with human like characteristics and artificial life, creating synthetic biology and synthetic psychology are other examples. While bioinspired design has been inspiring other disciplines, synthetic biology was inspired by synthetic chemistry and discipline of synthetic biology is interdisciplinary [35] Closer interaction of AI and transdisciplinary transdiscipline can be designed to benefit both.

AI will possibly create an artificial transdisciplinarity. AI has grumbled with some practical questions of philosophical concern like what makes a machine more intelligent the knowledge contains or the inference mechanism (thinking) it uses [36, 37] (Feigenbaum 1985 cited by Patterson 1990) and why does probability works [28]. AI combines human knowledge in unusual in ways breaking other disciplinary conventions by mixing knowledge of different organisms and disciplines [38]. This makes AI an innovative discipline.

Computing landscape consists of theory, practice and application continuum with SE at the center consisting of equal theory, practice and application [39]. Dawson et al. didn't include AI in continuum; AI would be placed in adjacent position to SE with some overlap. Comparison of a method from engineering design, SE and service design found that design is independent of domain [40]. Design approaches like design thinking and design science are used in science, engineering and social science disciplines. When an approach is used in several disciplines creating a transdisciplinary approach enables sharing of knowledge, experience and lessons learned between the disciplines. Computing Design science methods are transdisciplinary within computing disciplines and some beyond.

3.2 Need for Transdisciplinary Approaches

AI and SE lack general theories or discipline level theories, but they have many theory fragments and SE has recently developed specialized developed theories. Several specialized theories can be synthesized to create generalized theory. SE has theory fragments with potential to be developed into theories [41] also called micro-theories. Software engineers start developing novel software inventions and innovations any way they manage, knowledge experience gained is codified, systematized and then used to develop theories [42]. The general pattern of engineering theory development starts by creating concepts; that are then partially linked to create theory fragments and further development may create unified general theories. IEEE Software engineering body of knowledge (SWEBOK) [43] is a large collection of micro-theories [44] that are widely accepted by SE community. SE, SWEBOK and AI have no micro-theories for innovation and transdisciplinary indicating their innovation and transdisciplinarity is tacit. SWEBOK has micro theories for software design, management, process, construction, professional group dynamics and psychology, economics, and computing foundations: problem solving techniques, programming, programming languages, compiler construction, human factors; mathematical and engineering [43]. SWEBOK is most comprehensive source of software engineering knowledge.

There is increasing transdisciplinary research in information systems (IS). Kroeze et al. aligned information system with transdisciplinary perspective by reviewing transdisciplinary concepts relevant to IS and argued it would advance information systems. Multidimensionality of information system object is main tenet of transdisciplinary approach, that is acknowledged by several IS researchers [46]. Multidimensionality is key feature of computing disciplines including AI and SE especially because of their multidimensional knowledge. SE borrows from IS because of their shared similarity [47]. IEEE/ACM curriculum captures similarity of computing disciplines making it possible to extend Hall and Rapanotti assertion applies to all computing disciplines. AI is a computing discipline though it is not included as IEEE/ACM computing disciplines but one of its topics research methodologies is part of all IEEE/ACM computing disciplines.

The strong connection between computing disciplines including AI allow Hall and Rapanotti assertion of SE of borrowing from IS based on similarity and Kroeze et al. aligning IS and transdisciplinarity as way of advancing IS applicable to these computing disciplines especially AI that is transdisciplinary. Computing disciplines borrows and uses solutions from other disciplines and then develops a computing specific solution [48]. With islands of transdisciplinarity in AI, SE and IS computing research is ready to start carrying out transdisciplinary research including developing a computing trans-field.

In SE literature innovation theory based publications are rare and are rarer in AI. However recently design science [49, 50] and design thinking research and innovation methods [51, 52] that are innovation theory based are being introduced in SE but methods have not gained wide spread acceptance. Design science research frameworks have also been introduced in AI expert system development [53]. Design science and other research methods are used in information systems, computer science, information technology and computer engineering [48] these includes Artificial intelligence as being emphasized in computer engineering, computer science and information systems [43] while emphasis was low in Software engineering and IT in 2005 when research was published the emphasis has increased in information technology and SE as well as SE in AI. Aligning SE and AI research with design science and innovation methods will make two disciplines innovation and research activities more scientific [50]. In similar way study proposes aligning SE, AI and transdisciplinarity to aid in AI-SE transdisciplinary theory development. Transdisciplinary transdiscipline can provide innovation theory, help in integrating some SE and AI micro-theories towards creating a unified computing transdisciplinary theory.

SE avoids reinvention by reusing knowledge and some developed software artifacts in developing new software systems to reduce software cost and increase reliability. It also invests more resources to develop artifacts that can be reused in future products and services. Transdisciplinary is based on similar but broader idea of “develop once use everywhere” by developing transdisciplinary unified solutions to be ideally use by all disciplines, saving on efforts of reinventing solutions repeatedly in different disciplines. Transdisciplinary solutions provide promise of becoming more reliable than disciplinary solutions by being examined from more perspectives by potentially larger communities. Thus transdisciplinary knowledge unification as broader generalization of SE knowledge reuse.

Disciplinary knowledge has reached its limits with far reaching consequences for culture, science and social life, limits that can be overcome by transdisciplinarity [9]. SE and SE research and development create knowledge and technologies needed now and in future. Self-transcending knowledge transcends current and future knowledge [54]. Leveraging future knowledge for innovation involves tacit to self-transcending and self-transcending to tacit [55] transdisciplinary conversions. Two perspectives of transdisciplinary are Zurich that is based on mode 2 innovations and Nicolescuian that is based on quantum physics [56]. Software disciplines innovation is based on tacit mode 1 and tacit mode 2 because they have no explicit innovation and transdisciplinary theory. Mode1 is innovation within discipline, while mode 2 is transdisciplinary innovation involving knowledge and stakeholders in and beyond science and technology communities [57]. Mode 3 innovations combines mode 1, mode 2, national innovation system, sectorial innovation systems, cluster innovation system, triple helix, linear and nonlinear innovation models with people, culture, technology acting as glue that accelerates interaction between creativity and innovation [58] based on Nicolescu quantum duality. Design science research frameworks combine Mode 1 and mode 2 innovation to address problems with both research and practice dimensions. Mode 3 is transdisciplinary unifying innovation approaches and is suitable for new innovation movements like grassroots innovation.

SE has transdisciplinary foundations of systems, computing, management, engineering, economics, systems, cognition, linguistics and mathematics [59]. Mechatronics requires a transdisciplinary methodology to systematically unify knowledge within constituent disciplines [18]. Transdisciplinarity studies phenomena from unconventional angles, organizes knowledge beyond disciplines [60] and cross fertilizes different perspectives [61] in ways that increase creation and utilization of knowledge by different disciplines. Computing and engineering disciplines with significant tacit transdisciplinary characteristics are: artificial intelligence [8] mechatronics, Bioinspired design, and Nano technology. Other computing disciplines software engineering, information systems, information technology and computer engineering have components with some transdisciplinary tacit knowledge and other component have potential to benefit from acquiring transdisciplinary characteristics. Computing disciplines can learn from mechatronic experiences of introducing transdisciplinary methodology and from artificial intelligence tacit transdisciplinary. AI transdisciplinary tacit knowledge can be codified to make knowledge easy to leverage and adopt by other disciplines. According Science and engineering cycle codification of knowledge is a key step of science, engineering computing knowledge and experience development [42]. This key step has been skipped in transdisciplinary computing activities resulting slow computing transdisciplinary theory and experience development.

Design education needs to be infused with transdisciplinarity lens to prepare students for real world problem solving [62] with goal of making them transdisciplinary designers. This is necessary in developing and developed countries. Design education applies to design disciplines and disciplines with design fields like SE and AI. Introducing transdisciplinary lens or aligning transdisciplinarity are alternative ways towards increasing transdisciplinary orientation in disciplines. Technology following and catch-up entities need transdisciplinarity design to build their technology and innovation capabilities.

Systemism is driving a paradigm shift that aligns disciplines into an enduring transdisciplinary endeavor [63]. Systemism conceptualizes and organizes things, knowledge, theories and fields as systems [64]. Fields are parts of a discipline system that work together to achieve discipline's objective. Disciplines complement each other for sake of system of systems body of knowledge [63] similar to fields complementing each other for unity of discipline's body of knowledge. Transdisciplinarity aims to make best ideas from different disciplines sharable through creating transdisciplinary research models, frameworks and knowledge [10]. Transdisciplinary unification creates economy of scale as researchers from different disciplines share problems and create solutions together that benefits fields and disciplines.

Ontology is what exists and is the foundation of epistemology that deals what exists. Transdisciplinary emergentism is ontological super structure of epistemological integrationism [63]. Complex interaction of system parts causes emergence of system behavior. Knowledge across, between and beyond disciplines should interact in ways that result in emergence of desired properties. Transdisciplinary enables researchers to move beyond their disciplinary ontology and epistemology to be able to address complex problems through open ontology and epistemology that that allows high level knowledge synthesis [45]. Transdisciplinary is moving beyond at macro disciplines level, meso fields' level; micro ontology and micro epistemology levels.

Transformational social, political, technical, and economic and innovation knowledge, pluralism, norms and system knowledge [65, 66] interact in different ways determined by nature of problem to drive transdisciplinary research [66] and innovation by continuously discovering and learning how to integrate disciplines knowledge [67]. Transdisciplinary orientation corrects knowledge mismatches between academic production of knowledge and knowledge required to solve societal problems [68] by Exploring what is between intersections of people, knowledge, technology enables going between to go beyond [69] by using leveraging weak ties between them [70]. Weak ties present opportunities for discoveries that can lead to innovation.

Exploring unknown by going beyond can spark intergeneretivity thinking that engages multiple differences such as age, knowledge, experience, culture, indigenous and global connections to create new futures and discover new interconnectedness between human and non-humans [69] by leveraging social, knowledge, functional, cultural and organization distances between people [71]. Transdisciplinary Transdiscipline and biological genetics cross hybridize material across different disciplines or organisms to bear fruits in all of them [10]. Transdisciplinary allows access to a large collection of principles; methods and practices from different disciplines that design approaches combine in new ways to produce novel or improved solutions [72]. Hybridizing, exploring crossing, between and beyond are core transdisciplinary principles that are applicable to SE.

4 Towards a Transdisciplinarity AI-SE Trans-field

Tacit transdisciplinary knowledge SE developed artifacts are science and engineering cycle (SEC) [42], design patterns [8] and Unified Modeling Language (UML). SEC is transdisciplinary learning through doing, research and development computing, science, engineering and disciplines of artificial cycle AI being an engineering discipline uses this cycle. Exploratory programming [73] was first developed in AI but currently is being developed in SE and AI, while design science was developed in engineering. Both methods are transdisciplinary and are currently being developed in parallel in both disciplines. All these AI and SE artifacts are based on tacit knowledge and experience no explicit knowledge codified or transdisciplinary theory adopted by these disciplines. UML is being used in computing, engineering and business. People trained in more than one discipline are able to interpret different disciplinary languages [8] and assists transdisciplinary developers by being intermediaries between different disciplines. Some amount of transdisciplinary knowledge and experience is an added advantage when developing transdisciplinary artifacts.

Transdisciplinary universal methods are universal design theory [74], product development [75], product development cycle [76] created outside computing and designed for developing technologies in many disciplines including AI and SE. Among these methods only Savšek et al. makes explicit transdisciplinary theory. Ness 1997 proposed developing a transdisciplinary empirical engineering method for engineering and software engineering. These methods synthesize product development knowledge and transdisciplinary in a way that combines their strength in a way that increases productivity. Software engineers adopting and using these methods face the challenge of mastering disciplinary terminology and languages they are represented in. Transdisciplinary methods are less recognized because they use concepts not recognized in participating disciplines and require simultaneous combination of knowledge from several disciplines [75]. Other transdisciplinary methods are represented in disciplinary languages that are incomprehensible by disciplines that don't use the terminologies. This can be solved by standardizing and popularizing transdisciplinary language and representing all methods in a transdisciplinary language. Transdisciplinarity is a creative language that can help establish bridges between disciplines, fields and meanings [78]. The bridges make it easy to adopt and use of transdisciplinary knowledge and methods by participating disciplines.

Artificial intelligence (AI) transdisciplinary basic studies and research projects have led to ground breaking theories and technology innovations [79], due to AI being transdisciplinary, bioinspired design [8] and Meta transdisciplinary discipline. AI product and service designers are transdisciplinarians who synthesize human knowledge from different disciplines in ways not constrained by disciplinary boundaries with results observed to perplex philosophers and linguists [38]. SE transforms human knowledge into executable code [8] that's incapable of reasoning. AI on other hand transforms human knowledge into executable code capable of reasoning with program stored knowledge. AI uses cognitive brain model nature laws to bench mark what it can do and or not do [79] and Turing machine model. Other AI biological models are genetic model and ant colony model. AI has many years of experience codifying human explicit and tacit expertise in building expert systems. This experience would aid in codifying knowledge in the two engineering disciplines.

Design practice has transcended different design boundaries to become transdisciplinary which has transformed conventional product development from mono-disciplinary to highly transdisciplinary and designers from different disciplines perform similar tasks but use different terminologies [80]. A transdisciplinary terminology can make collaboration of designers from different discipline easier. AI and SE designers have explicit and tacit disciplinary knowledge and tacit transdisciplinary knowledge. Providing SE and AI designers with transdisciplinary theory, explicit knowledge and methods would provide means for them to leverage transdisciplinarity. Most craft technology developers are not designers but are likely to significantly benefit most from creation of a transdisciplinary environment.

Intelligent SE is a new field with two goals creating intelligent solutions for SE and creating SE solutions for AI [13]. Complementarity between AI and SE presents opportunity for hybridizing AI techniques and SE processes [81]. Transdisciplinary innovation and methods accelerate divergent thinking. Transdisciplinarity improves creating minimum viable products by software startups through enabling them navigate challenges [81] like lack of capital, manpower and saleable product. Single owner software startups have to be transdisciplinary since the owner performs all functions in relevant disciplines of: SE, innovation, business and accounting best handled by unifying knowledge.

Training software engineers to be AI engineers is viable and there is need of AI engineers that combine SE skills and machine learning skills, who can be trained by building AI skills on their software engineering education foundations [11]. AI engineering and SE have same research agenda to develop tools, techniques, frameworks and best practices that practitioners use to build production ready systems [82]. Two disciplines research agenda can use analogy to converge their interests in AI-SE trans-field. SE agile software development methods can be adapted by orienting them towards exploration and research to develop AI systems [83]. AI based systems are software systems with SE and at least one AI component are becoming pervasive in society; however there is limited synthesized SE knowledge for building such systems [14]. AI and SE have been intertwined since their beginning with tools of each discipline being used in other, with increasing application of SE techniques in AI and vice versa for example intelligent techniques in agile software development [84]. SE community exploits emerging areas of research and practice through creating synergies between AI and SE [85]. Software engineers are concerned with complexity while machine learning systems developers are concerned with lack of knowledge of architecture and design patterns [86]. The problem of each discipline has solved by other, but solutions are difficult to use because they are not transdisciplinary. Organizations use internal and external knowledge to determine intelligent techniques to use in developing ISE solutions [87]. Transforming knowledge to products requires crossing organizational and disciplinary boundaries.

Technical singularity is hypothesized as situation where AI self-improving programs would cause intelligence explosion that yields intelligence exceeding [88] (Nicolescu 2017:155). While currently this hypothesis for design fiction movies, equivalent to accepting self-driving cars in beginnings of AI, some seeming impossible hypothesis, sometimes get to become intermediate impossible as it seems now with technology singularity, if AI improvement continues at same rate, it could become reality. The AI-SE trans-field research should have following transdisciplinary characteristics: shared concepts, tools and methods to solve semi structured and ill structured research problems, build bridges between natural science, social science, humanities and Engineering and enable strong collaboration between AI and SE disciplines and beyond [89]. Bridges would aid knowledge and solutions flow between these disciplines which is catalyst for innovation.

Scope and boundaries of disciplines are set based by limits of human professional life span, human memory capacity and brain processing capability. These determine what a professional can learn and practice effectively in one's community of knowledge in a life time.

5 Conclusion and Discussion

Literature review was used to identify transdisciplinarity software engineering and artificial intelligence fields, methods, elements, discover how SE and AI can be aligned with transdisciplinary and an AI-SE trans-field specializing in transdisciplinarity research and theory development. Several software fields and field fragments were identified including one at intersection of AI and SE: intelligent SE that has merged with search based SE. Transdisciplinary methods applicable to software development were identified both within AI and SE and those developed from other disciplines. The challenge of adopting and using transdisciplinary software methods developed outside the two disciplines is dealing in disciplinary language representation that is hindrance to adoption. Transdisciplinary language representation can make these methods easier to adopt by AI and SE engineers. Transdisciplinary language would ensure learning all transdisciplinary methods require learning transdisciplinary language once and then when adopting other methods adopter will method specific issues. One problem of AI and SE transdisciplinarity artifacts including methods is lack theory and codification of innovation and transdisciplinary knowledge. Transdisciplinary theory can support mode 2 and mode 3 software disciplines innovations and aid their maturing as engineering disciplines.

The main goal of transdisciplinarity is to unify knowledge which makes constituent knowledge complementary. Aligning AI and SE with transdisciplinarity makes it easier to use transdisciplinary knowledge within the disciplines, develop their intersections and some transdisciplinary knowledge internally. AI can contribute to computing and engineering transdisciplinarity development because its foundations are intelligence and experience of mixing knowledge from biology and psychology and other disciplines validated through building machines that work.

Mechatronics has transcending identity [18] it interacts, overlaps with SE. Both are based on transcending identity aiding sharing their transdisciplinary experiences. Creating the AI-SE trans-field would be a first step towards creating a computing transdisciplinary trans-field. This fits within computing disciplines pattern of first adopting and using an approach borrowed from outside computing and then creating a computing approach [48]. Because of interrelated nature of SE, information systems and computer science they should share knowledge [30] through transdisciplinarity. Interrelatedness extends to artificial intelligence and other computing disciplines.

Most SE research is not usable in industrial practice [90] and they is need for philosophy of SE interdisciplinary discourse between philosophically minded software engineers, other engineers, professional philosophers and social scientists [91]. Transdisciplinary philosophy can explore foundations of SE and AI.

Funding: This research article received no external funding.
Conflicts of Interest: The author declares no conflict of interest.

References

[1] Zahra, S. A. & Newey, L. R. (2009). Maximizing the Impact of Organization Science: Theory-Building at the Intersection of Disciplines and/or Fields. *Journal of Management Studies*, 46(6), 1059-1075.

[2] Fagerberg, J. Verspagen, B. (2009). Innovation studies — The emerging structure of a new scientific field, *Research Policy*, 38, 218–233.

[3] Van Zyl, I. (2015). Disciplinary Kingdoms: Navigating the Politics of Research Philosophy in the Information Systems, EJISDC, 70(1), 1-17.

[4] Pohl, C. (2010). From Transdisciplinarity to Transdisciplinary Research. *Transdisciplinary Journal of Engineering & Science,* 1, 65-73.

[5] Augsburg, T. (2014). Becoming Transdisciplinary: The Emergence of the Transdisciplinary Individual, *World Futures,* 70:3-4, 233-247, DOI: 10.1080/02604027.2014.934639

[6] Linkman, S. (1997). Experimentation as a vehicle for software technology transfer- A family of Software Reading Techniques. *Information and Software Technology,* 39.

[7] Rogers, E. M. (1995). Diffusion of Innovations, Free Press.

[8] Korecki, S. (2008). Inspired Design: Using Interdisciplinary and Biomimicry for Software Innovation,

[9] Nicolescu, B. (2010). Methodology of Transdisciplinarity: Levels of Reality, Logic of the Included Middle and Complexity. *Transdisciplinary Journal of Science and Engineering*, 1(1), 19-38.

[10] Cohen, E., & Lloyd, S. (2014). Disciplinary Evolution and the Rise of the Transdiscipline. *Informing Science: the International Journal of an Emerging Transdiscipline,* 17,189-215.

[11] Heck, P. Schouten, G. (2021). Turning Software Engineers into AI Engineers, arXiv.

[12] Xie, T. (2013). The synergy of human and artificial intelligence in software engineering. In Workshop

[13] Xie, T. (2018). Intelligent Software Engineering: Synergy between AI and Software Engineering, ISEC'18, ACM Hyerabad, India.

[14] Martínez-Fernández, S. Bogner, J. Franch, X. Oriol, M. Siebert, J. Trendowicz, A. Vollmer, A. Wagner, S. (2021). Software Engineering for AI-Based Systems: A Survey, arXiv.

[15] McDermott, T. DeLaurentis, D, Beling, P. Blackburn, M. & Bone, M. (2020). AI4SE and SE4AI: A Research Roadmap, Insight, 23(1), 8-13.

[16] Law, E. (2011). Defining (Human) Computation, ACM.

[17] McGregor, S. L. T. (2013). Transdisciplinarity and Biomimicry, Transdisciplinary Journal of Engineering & Science, 4, 57-65.

[18] Berian, S. & Maties, V. (2011). A Transdisciplinary Approach to Mechatronics. *Transdisciplinary Journal of Engineering & Science,* 2, 20-32.

[19] Rose, J. (2010). Software Innovation: Eight Work-Style Heuristics for Creative System Developers, Software Innovation, Advances in Software Engineering, Aalborg University.

[20] Tate, D. (2010). Designing Transdisciplinary Discovery and Innovation: Models and Tools for Dynamic Knowledge Integration. *Transdisciplinary Journal of Engineering & Science,*1(1), 105-124.

[21] Fischer, G. & Scharff, E. (2000). *Meta-Design: Design for Designers.* DIS '00, Brooklyn, New York. ACM.

[22] Hohpe, G. Ozkaya, I. Zdun, U. & Zimmermann, O. (2016). The software architect's role in digital age, *IEEE Software.*

[23] Hoffmann-Riem, H. (2008). *Idea of the Handbook*, in eds. Hadorn et al. Handbook of transdisciplinary research, Springer.

[24] Ertas, A. Tanik, M. M. Maxwell, T. T. (2000). Transdisciplinary Engineering Education and Research Model, *Transactions of the SDPS*, 4(4), 1-11.

[25] Humphrey, W. S. (1995). *A discipline of Software Engineering.* Addison-Wesley publishing company.

[26] Glass, R. L. Vessey, I. & Ramesh. V. (2002). Research in Software Engineering: An Analysis of the Literature. *Journal of Information and Software Technology,* 44(8), 491-506.

[27] Poole. D. L. Mackworth, A. K. (2010). *Artificial Intelligence: Foundations of Computational Agents.* Cambridge University Press.

[28] Russel, S. Norvig, P. (2010). *Artificial intelligence a modern approach.* Prentice-Hall.

[29] Brey, P. & Søraker, J. (2009). *Philosophy of Computing and Information Technology Philosophy of Technology and Engineering Sciences*, in the Handbook for Philosophy of Science. Eds. Meijers A., Elsevier, 14.

[30] Vessey, I. Ramesh, V. & Glass, R.L. (2005). A Unified Classification System for Research in the Computing Disciplines. *Information and Software Technology*, 47(4), 245-255.

[31] Harman, M. (2011). Software Engineering Meets Evolutionary Computation, IEEE.

[32] Mann, D. (2004). TRIZ for Software, *TRIZ Journal.*

[33] Van Tillaart, R. (2006). TRIZ and Software - 40 Principle Analogies, a Sequel, *TRIZ Journal.*

[34] Haugeland, J. (1996). What Is Mind Design? in Mind design: Philosophy Psychology Artificial Intelligence eds. Haugeland, J. & Book, B. A., Massachusetts Institute of Technology.

[35] Malinova, V. Nallani, M. Meier, W.P. Sinner E.K. (2012). Synthetic biology, inspired by synthetic chemistry, *FEBS Letters,* 586, 2146–2156.

[36] Feigenbaum, E. A. (1985). Paper presented at five generation joint conference on artificial intelligence, Los Angeles, CA.

[37] Patterson, D. (1990). *Artificial intelligence and expert system*, Prentice-Hall.

[38] Jackson, P. (1999). *Introduction to Expert Systems,* 3rd edition, Addison-Wesley.

[39] Dawson, W. C. (2015). *Projects in computing and information systems: Students guide*, Pearson 3rd Edition.

[40] Kannengiesser, U. Gero, J. S. (2015). Is designing independent of design? Comparing models of engineering, software and service design, *Res Eng Design.*

[41] Stol, K. & Fitzgerald, B. (2013). Uncovering Theories in Software Engineering, IEEE.

[42] Shaw, M. & Garlan, D. (1996). Software Architecture: Perspectives on an Emerging Discipline, Prentice-Hall.

[43] Bourque, P. & Fairley, R. E. (2014). Guide to the Software Engineering Body of Knowledge, IEEE.

[44] Allen, B. A. (2015). Design artifacts are central to foundations of a theory of software engineering , Technical report MSU-20150420.

[45] Kroeze, J. H. Travica, B.& Zyl, I. V. (2019). Information Systems in a Transdisciplinary Perspective: Leaping to a Larger Stage, *Alternation Special Edition*, 24, 9-47, https://doi.org/10.29086/2519-5476/2019/sp24.2a2

[46] De Albuquerque, J. & Simon, E. (2007). Dealing with Socio-Technical Complexity: Towards a Transdisciplinary Approach to information systems research. European Conference on Information Systems.

[47] Hall, J. G. & Rapanotti. L. (2017). A design theory for software engineering, *Information and software technology*, 87.

[48] Holz, H. J. Applin, A. Haberman, B. Joyce, D. Purchase, H. & Reed, C. (2006). Research Methods in Computing: ITiCSE'06, ACM, 96-114.

[49] Wieringa, R. J. (2014). Design Science Methodology in Information System and Software Engineering, Springer Verlag.

[50] Engstrom, E. Storey, M. Runeson, P. Host M. Maria Teresa Baldassarre T. M. (2020). How software engineering research aligns with design science: A review, Empirical Software Engineering, 25, 2630–2660.

[51] Lindberg, T., Rauth, I., Köppen, E. & Meinel, C. (2001). Design Thinking in the IT Industry: Exploring Language Games on Understanding, Implementation and Adoption. Published in the proceedings of Design Thinking Research Symposium.

[52] Pereira, L. Parizi, R. Marczak, S. & Conte T. (2021). Design Thinking Techniques Selection in Software Development: On the Understanding of Designers and Software, *Engineers Choices*, ICEIS 2021, 2, 353-360.

[53] Miah, S. J. & Genemo, H. (2016). A Design Science Research Methodology for Expert Systems Development, *Australasian Journal of Information Systems,* 20.

[54] Scharmer, C. O. (2001). Self-transcending Knowledge: Organizing around Emerging Realities, Managing Industrial Knowledge: Creation, Transfer and Utilization, 1, 68–90.

[55] Harmaakorpi, V. & Melkas, H. (2005). Knowledge Management in Regional Innovation Networks: The Case of Lahti, Finland, *European Planning Studies,* 13(5). 641-659.

[56] McGregor, S. L. T. (2017). The Nicolescuian and Zurich Approaches to Transdisciplinary,*Integral Leadership Review.*

[57] Gibbons, M. Limoges, C. Nowotny, H. Schwartzman, S. Scott, P. & Trow, M. (1994). The New Production of Knowledge: The Dynamics of Science and Research in Contemporary Societies. Sage.

[58] Carayannis, E. G.& Campbell, D. F. (2009). 'Mode 3' and 'Quadruple Helix': Toward a 21st Century Fractal Innovation Ecosystem, *Int. J. Technology Management,* 46(3/4).

[59] Uysal, M. P. (2016). In Search of Software Engineering Foundations: A Theoretical and Trans-disciplinary Perspective,*IJCTE*, 8(4), 328- 331.

[60] Bernstein, J. H. (2015). Transdisciplinarity: A review origins, development and current status, *Journal of research practice,* 11(1).

[61] Cummings, S. Regeer B. Ho W. & Zweekhorst M. (2013). Proposing a Fifth Generation Knowledge Management for Development: Investigating Convergence between KM4D and Transdisciplinary Research, *Knowledge Management for Development Journal,* 9(2).

[62] Ambole, A. (2020). Embedding Design in Transdisciplinary Research: Perspectives from Urban Africa, *Design Issues,* 36(2), 28-39.

[63] Hofkirchner, W. (2017). Transdisciplinarity Needs Systemism, *Systems,* 5(15), 1-11.

[64] Bunge, M. (2000). Systemism: The Alternative to Individualism and Holism, *Journal of Socio-Economics*, 29, 147–157.

[65] Pohl, C. & Hadorn, H. G. (2007). Principles for Designing Transdisciplinary Research, Proposed by the Swiss Academies of Arts and Sciences.

[66] Hadorn, G. H. Biber-Klemm, S. Grossenbacher-Mansuy, W. Hoffmann-Riem, H. Joye, D. Pohl, C. Wiesmann U. & Zemp, E. (2008). The Emergence of Transdisciplinarity as a Form of Research, in eds. Hadorn et al. Handbook of transdisciplinary research, Springer.

[67] Zafeirakopoulos, M. & Bijl-Brouwer, M. (2018). Exploring the Transdisciplinary Learning Experiences of Innovation Professionals, *Technology Innovation Management Review*, 8(8), 50-59.

[68] Pohl, C. & Hadorn G. H. (2008). Core Terms in Transdisciplinary Research, in eds. Hadorn et al. Handbook of transdisciplinary research, Springer.

[69] Whitehouse, P. J. Bodiford, K. Standar, P. Aarali A. M. Asiimwe, S. Vegter, V. Xi, W. & Torres-Davila, P. (2021). Intergenerative Transdisciplinarity in Glocal Learning and Collaboration, *Transdisciplinary Journal of Engineering & Science,* 12, 29-37.

[70] Burt, R. S. (2004). Structural holes and good ideas, *American Journal of Sociology,* 110(2), 349–399.

[71] Parjanen, S. (2008). Distance and Proximity as Source of Innovation in Regional Innovation Processes, Conference on Regional Development and Innovation Processes, Porvoo, Finland.

[72] Dorst, K. (2018). Mixing Practices to Create Transdisciplinary Innovation: A Design-Based Approach, *Technology Innovation Management Review,* 8(8), 60-65.

[73] Kerry M. B. Myers B. A. (2017). Exploring Exploratory programming, IEEE symposium on visual languages and human centric computing.

[74] Grabowski, H. Lossack, R.S. & El-Mejbri, E.F. (1999). Towards a Universal Design Theory, in Integration of Process Knowledge into Design Support Systems, Springer, 47-56.

[75] Savšek, T. Makovec, I. & Cerovšek, M. (2015). Transdisciplinary Product Development, *Journal of Universal Excellence,* 4, 171–182.

[76] Gumus, B. Ertas, A. Tate, D. & Cicek I. (2008). The Transdisciplinary Product Development Lifecycle model, *Journal of Engineering Design,* 19(3) 185-200, DOI:10.1080/09544820701232436.

[77] Ness, P. E. (1997). Creative Software development: An empirical modeling framework, PhD thesis , university of Warwick.

[78] Pop, I. G. (2009). Creativity as a reflexive transdisciplinary language in the mechatronical knowledge, *Problems of Education in the 21st Century,* 13, 94-102.

[79] Wang, Y. & Tunstel, E. (2019). The Emergence of Abstract Sciences and Transdisciplinary Advances, *IEEE Systems, Man, & Cybernetics Magazine*, 12-19.

[80] Sharunova, A. Butt, M. Kresta, S. Carey J. Wyard-Scott, Adeeb S. Blessing, L. & Qureshi, A. J. (2017). Cognition and Transdisciplinary Design: an Educational Framework for Undergraduate Engineering Design Curriculum Development, Proceeding 2017 CEEA17 Conference.

[81] Shehab, M. Abualigah, L. Jarrahc, M. I. Alomarid, S. A. & Daoude, M. (2020). Artificial intelligence in software engineering and inverse: review, *International Journal of Computer Integrated Manufacturing,* 1-16.

[82] Heck, P. Schouten, G. (2021). Lessons Learned from Educating AI Engineers, arXiv.

[83] Singla, K. Bose, J. Naik C. (2018). Analysis of Software Engineering for Agile Machine Learning Projects, IEEE.

[84] Perkusich, M. e Silva, L. Costa, A. Felipe Ramos, F. Saraiva, R. Freire, A. Dilorenzo, E. Dantas, E. Santos, D. Gorgonio, K. Almeida, H. Perkusich, A. (2019). Intelligent Software Engineering in the Context of Agile Software Development: a Systematic Literature Review, *Information and Software Technology.*

[85] Holland, C. Tanenbaum J. (2020). Emerging Technologies 2020: Six Areas of Opportunity, Software Engineering Institute.

[86] Washizaki, H. Uchida, H. Khomh, F. Gueheneuc, Y. (2019). Studying Software Engineering Patterns for Designing Machine Learning Systems, IEEE.

[87] Chagas, J. Silva, L. Perkusich, M. Neto, A. Albuquerque, D. Valadares, D. C. Almeida H. Perkusich, A. (2020). On the Reuse of Knowledge to Develop Intelligent Software Engineering Solutions, SEKE, 101-106.

[88] Nicolescu, B. (2017). Technological Singularity: The Dark Side, in Transdisciplinary in Higher Education Eds. Gibbs P., Springer.

[89] Ertas, A. (2012). *Engineering Mechanics and Design Applications: Transdisciplinary Engineering Fundamentals,* CRC Press Taylor& Francis Group.

[90] Rombach, D. & Seelisch, F. (2008). Formalisms in Software Engineering: Myths versus Empirical Facts. Lecture Notes in Computer Science, 5082, 13-25, Springer-Verlag.

[91] Gruner, S. (2011). Problems for a philosophy of software engineering, *Minds and Machines,* 21(2), 275-299.

About the Author

Wangai Njoroge Mambo started his career as systems analyst/programmer after that has been computing lecturer for two decades in several Kenya universities. He obtained a masters in computer applications from Zhejiang University and BSc (Chem and Math's) degree from university of Nairobi Kenya. Currently he is an adjunct lecturer at United States International University Africa. His research interests are software engineering, artificial intelligence innovation and transdisciplinary product development.