ERC Advanced Grant 2017
Research proposal [Part B1]

White-Box Self-Programming Mechanisms

WhiSeMech

Cover page:

– Principal investigator (PI): **Giuseppe De Giacomo**
– Host institution: **Università degli Studi di Roma "La Sapienza"**
– Proposal duration: **60 months**

We are witnessing an increasing availability of **mechanisms** that offer some form of programmability. These include software, manufacturing devices, smart objects and smart spaces, intelligent robots, business process models, component-based systems, and many others. All these mechanisms have some built-in capabilities that give them a dynamic behavior, which can be organized, refined and repurposed through programming.
WhiSeMech aims at allowing such mechanisms to **program themselves**, without human intervention. Through this self-programming ability such mechanisms can tailor their behavior so as to achieve desired goals, maintain themselves within a safe boundary in a changing environment, and keep following rules, regulations and conventions that evolve over time.
Though, unlike some machine learning approaches, WhiSeMech aims at the self-programming mechanisms that are **white-box**: specifications and automatically synthesized programs must be human comprehensible. In other words, WhiSeMech aims at obtaining self-programming mechanisms whose behavior is fully **explainable in human terms** by design.
WhiSeMech's scientific work will merge key ideas from **Reasoning about Action** in *Knowledge Representation* on how to represent the domain of interest, the system, and their properties in a high-level human comprehensible fashion; from **Data-aware Processes** in *Databases*, on how to build data-aware dynamic behaviors; from **Verification and Synthesis** in *Formal Methods* on providing mathematically elegant foundations for synthesis, though focusing on computationally more tractable formalisms recently proposed in Reasoning about Action; and from **Planning** in **Artificial Intelligence** to gain algorithmic insight to the synthesis process.
WhiSeMech grounds its scientific results on diverse real **application contexts**, including manufacturing systems (Industry 4.0), smart spaces (IoT) and business processes (BPM).

a Extended Synopsis of the scientific proposal

LONG TERM VISION

Consider the following scenario.

> *After a long week-end, the human supervisor inspect the manufacturing system and notice that a production line has significantly slowed down, though it is still producing. She queries the system on what it is doing. The system expose the revised process, which is avoiding the use of the production island 176-671, by repurposing the tools in island 176-716 and sending pieces there. She then queries why the system has reprogrammed itself to do so. The system answer by showing that on Sunday 11:43pm the island 176-671 started to producing an unacceptable percentage of defective pieces, based on tests performed during production. So following the recovery specification that asked minimize cost while keeping the process within safe boundary, instead of shutting down the production line, it analyzed the available capabilities and reprogrammed itself to perform the current revised process moving the fabrication of the pieces to island 176-716 and reconfiguring the tools there to do so. On request, the data about the defective process are displayed to the supervisor and stored for planning maintenance. Moreover the system provides formal evidence that the reconfigured line meets both the product specifications and the system constraints.*

In the senario above we have a **mechanism** (the manufacturing system) with some built-in capabilities to act and react to changes (making defective pieces), which enacts a *dynamic behaviour* (the production process) to meet its *specifications* (constraints on the product and production model). Crucially, the mechanism has **self-programming** capabilities that can use to modify its current behavior *without human intervention*. Notably the mechanism can be *queried* to expose, in terms understandable to humans, its *self-synthesized program*, the *specifications* used and the *relationship* between the specifications and the synthesized program. In a slogan the mechanism is **white-box**.

The overarching objective of WhiSeMech is to make this vision a reality:

> WhiSeMech *aims at laying the theoretical foundations and practical methodologies of a science and engineering of* ***white-box self-programming mechanisms****.*

To make apparent the significance of this enterprise, WhiSeMech will ground its research in three contexts currently considered of pivotal importance in Business, namely:

1. **Manufacturing**, where a significant research efforts are focusing on improving flexibility, agility and productivity of manufacturing systems, under the umbrella term *Industry 4.0*, or *4th industrial revolution*.[1]
2. **Internet of Things**, which is rising as a virtual fabric that connects "things" equipped with chips, sensors and actuators and allows for building **smart objects** and **smart spaces** with high level of awareness of the environment and its human occupants.[2]
3. **Business Process Management**, which advocates explicit conceptual descriptions of a process to be enacted within an organization or possibly across organizations, and which is instrumental to business processes improvement, the top business strategy of CIOs in organization according to Gartner. [3]

Interestingly forms of self-programmability have been advocated in all the above contexts. For example, it is advocated that cyber-physical systems in Manufacturing or Internet of Things should be able to **adapt** themselves to current users and environment by **exploiting information gathered at runtime**. However it is considered **impossible to determine a priori all possible adaptations** that may be needed: self-programming abilities would be highly disederable [74]. In Business Processes, it is considered important for the next generation of process management systems to allow processes to automatically **recover executions** when unanticipated exceptions occur, without explicitly defining

[1] M. Lorenz et al. *Man and Machine in Industry 4.0: How Will Technology Transform the Industrial Workforce Through 2025?* The Boston Consulting Group. 2015.
[2] C. MacGillivray et al. *Worldwide Internet of Things Forecast Update, 2016-2020.* IDC. Doc # US40755516. 2016.
[3] Gartner Group. *BPM Survey Insights.* Gartner Report. http://www.gartner.com/it/page.jsp?id=1740414.

a priori **recovery policies**, and **without the intervention of domain experts** at runtime. These self-programming abilities would reduce costly and error-prone manual ad-hoc changes, and would relieve software engineers from mundane adaptation tasks [58]. Note that some of these concerns have been shared by **autonomic computing**, which has promoted self-configuration, self-healing, self-optimization, and self-protection, though by using policies provided by IT professionals [54].

Although the interest is clearly apparent, currently these self-programing abilities are missing in actual mechanisms, and science is focussing on limited forms of self-programming, e.g., for exception handling and recovery, or forms of composition and autonomic reconfiguration.

WhiSeMech instead will consider self-programmable mechanisms, as forms of **Agents** studied in **Artificial Intelligence** [68, 85]. [4]

More precisely, WhiSeMech intends to make a quantum leap in mechanisms' self-programming abilities while keeping them white-box. To do so, WhiSeMech will push forward the emerging cross-fertilization among from **Reasoning About Action** in **Knowledge Representation**, **Data-aware Processes** in **Databases**, **Verification and Synthesis** in **Formal Methods** and **Planning** in **Artificial Intelligence**.

> *The PI has profoundly contributed to all these areas, and he is one of the most prominent AI scientist leading this cross-fertilization.*

Through enhanced self-programming abilities such mechanisms can, e.g.:

- *Achieve desired goals*, that is guarantee that a certain desired state of affair is eventually reached. In the above example a manufacturing system automatically reconfigures the fabrication process if a some tool is producing too many defective pieces, by changing the sequencing of processing units so as to momentarily cut-out the defective tool from the process.
- *Maintain themselves within a safe boundary* in the changing environment in which they operate. For example a smart space system may keep the desired temperature and humidity in a museum room at some desired level, even in presence of a particularly large crowd of visitors, possibly by momentarily repurposing other actuators, such as the general public air conditioning system.
- *Keep following rules, regulations and conventions* that evolve over time while enacting their behavior. For example, to answer a new privacy regulation, a business process may refine its behavior to guarantee that the sensible data are eventually erased from the system before the completion of each process instance.

More generally, WhiSeMech wants to enable mechanisms to act in an informed and intelligent way in their environment, by changing the way they behave as a consequence of the information they acquire from the external world, and they exchange with the humans operating therein.

Since "*with great power comes great responsibility*", introducing advanced forms of self-programming calls for the ability to make the behavior automatically synthesized by the mechanism **understandable** to human supervisors. So it is indeed crucial to develop self-programming mechanisms that are **white-box**: in every moment the mechanism can be queried for its specifications, its behavior and how it relates to the specifications. Ultimately it is the possibility of **explain in human terms** the resulting behavior that will make white-box self-programming mechanisms **trustworthy** [21, 62]. Being white-box contrasts with most current approaches, which consider acceptable synthesized solutions that remain opaque to humans, as long as they work [59, 77].

In the first example above, both the reconfiguration goal (cutting out a defective tool) and how the fabrication process has been modified need to be explicitly understandable by the humans analyzing the manufacturing system. In the second example, the sudden repurposing on the air conditioning system also needs to be understandable to humans as a reaction to avoid violating certain safety conditions. Similarly, in the third example, the goal of erasing sensible data from the system, and even more importantly how this is achieved, must be understandable.

We further stress that the need to move towards **white-box** approaches is advocated by a large part

[4] We stress that WhiSeMech does not aim at general AI, but envisions self-programming mechanisms that act intelligently within the specific domain of interest in which they operate.

of the **AI community** [69], and has been recently taken up by DARPA within the context of machine learning, through the DARPA-BAA-16-53 "Explainable Artificial Intelligence (XAI)" program[5]

> *Knowledge representation, the primary field of the PI, will be central for realizing the shift towards a white-box approach.*

As a result, WHISEMECH is **very timely** and of **greatest significance for European science**.

OBJECTIVES

Towards the goal of building **white-box self-programming mechanisms**, WHISEMECH will address the following objectives.

1. **Equip mechanisms with general self-programming abilities.** Mechanisms need general self-programming abilities, not restricted to a particular task, such as exception recovery, but ready to refine and modify the behavior of the mechanisms as new opportunities or constraints arise. In other words, we need to aim at advanced forms of **process synthesis** as those studied in **reactive synthesis**. Over the years the Verification and Synthesis community in Formal Methods has developed a comprehensive and mathematical elegant theory of **reactive synthesis** [65]. Such theory however has not yet found broad practical application because of the **intrinsic difficulties** of certain algorithms and constructions [82]. WHISEMECH aims at **sidestepping** these notorious difficulties altogether by focusing on non-traditional forms of specification formalisms, such as LTL and LDL on finite traces, recently proposed in **reasoning about action** in AI [81, 41, 42, 20] and in **declarative business processes** [83].

2. **Make self-programming abilities available while in operation.** Self-programming abilities are needed while mechanisms are in operation, that is while the mechanisms are executing, not just at design time. We expect self-programming mechanisms to be able to reprogram themselves under new acquired information or changes in the specifications, while already in operation. This deeply relates self-programming to **Planning** in AI [48, 49, 61]. In particular like agents in planning, we expect mechanisms to be able to handle quickly and efficiently most cases, i.e., those cases that do not require to solve difficult, "puzzle-like", situations. Indeed while the Planning community has concentrated on simpler forms of process synthesis, it has developed a sort of science of search algorithms for planning, which has brought about improvements by orders of magnitude in the last decade. [67, 80, 56, 37] WHISEMECH will exploit this knowledge and extend it to generalized forms of planning and to reactive synthesis.

3. **Make white-box self-programming mechanisms verifiable.** WHISEMECH aims at building self-programming mechanisms whose replanned behaviors are *verifiable* agains their specifications, e.g., by **model checking**, possibly **modulo theories**. This is a crucial step towards the understandability required by a **white-box approach**. In this way mechanisms can be checked, e.g., to understand if important safety conditions are satisfied. In tackling this aspect WHISEMECH will also leverage on recent advances of model checking of autonomous agents [57].

4. **Allow learning and stochastic decisions, while remaining within safe bounds.** We want to allow mechanisms to have forms of decision making that resist formal analysis (at least in human terms), because we want to make use of the possibilities that advancements in deep learning, MDPs, and reinforcement learning bring about. Though, while the actual execution could be chosen stochastically, we do want to have guarantees on **all possible generated executions**. In this way, it is the **entire space of solutions** that has **formal guarantees**, and the specific solution chosen by the learning algorithm or the stochastic decision maker will also satisfy them. This calls for allowing **coexistence of logical constraints with stochastic solutions**, a theme that has only be scratched by the scientific community so far [5, 79, 12]. We also observe that synthesis against constraints has been used to bound the possible solutions in several context, most notably in supervisory control [84, 2].

5. **Make white-box self-programming mechanisms comprehensible to humans**. WHISEMECH requires specifications, the space of solutions and the relationship between solutions and

[5] http://www.darpa.mil/program/explainable-artificial-intelligence

specifications to be **comprehensible to humans**. Neural networks, can be effectively used for finding a specific solution, but they cannot be used as a human comprehensible representation of the solution space, since we do not have control on the abstraction/compression they perform [60]. This means that specifications and solution spaces must be **semantically** described at high-level using predicates that are understandable to humans, as advocated by **Knowledge Representation** in AI (that is, it is fine to say `Island 176-671 under stress`, but not to say `Flag123456=on`) [4, 46, 13, 76, 55].

6. **Make self-programming mechanisms data-aware.** During the execution, new facts about the world are observed, learned, or received as input. This calls for a representation that distinguishes **intensional information** such as that provided by knowledge on the domain, from **extensional information** provided by actual data. Self-programming mechanisms leverage on the intensional information to be able to interpret new data (extensional information) acquired, observed, learned. Notice that this calls for a **relational (first-order) representation of the state**. New results on verifiability of **data-aware processes**, based on faithfull abstraction fo finite state transition systems, are crucial[22, 1, 23, 53, 6, 31, 17].

7. **Favor component-based approaches.** By no means we should consider programmable mechanisms to be formed by a single unit only. **Service-Oriented Computing** and **Open APIs** frameworks are long pushing for **component-based systems**, in which a set of components are **customized** and **orchestrated** to deliver a required service [11]. Indeed, understandability calls for building **high-level components** that are relatively simple to understand, verify and combine. Then, it is crucial to study how **composing** "correct" components leads to an overall "correct" behavior. WhiSeMech will leverage on the body of work on composition and customization developed in SOC and more recently in AI [78, 8, 39, 63, 28]. Moreover, execution should be **monitored** so that in case of failure it is possible to identify the responsible component, and recover the situation by reprogramming the mechanisms for alternative solutions that circumvent the failing component [24, 58].

METHODOLOGY

WhiSeMech lies at the intersection (and crosses the boundaries) of four vibrant sub-fields of contemporary computer science research: **Reasoning About Action**, **Data-aware Processes**, **Verification and Synthesis** and **Planning**.

Reasoning About Action. WhiSeMech will leverage on the large body of work on reasoning about action developed in Knowledge Representation, to which the PI has contributed significantly in the years [40, 29, 36, 72, 71, 35, 30, 33, 34, 3]. From such work, WhiSeMech will draw key ideas on how to represent mechanisms, the domain in which they are operating, and the properties of interest in a high level human comprehensible fashion. However particular attention will be given to computational effectiveness, in line with some recent exploratory work by the PI [31, 32, 17]. On this theme, the PI team will collaborate with Yves Lesperance (York U., Toronto, Canada), Hector Levesque (U. Toronto, Canada), Sebastian Sardina (RMIT, Melbourne, Australia), and Yongmei Liu (Sun Yat-sen U., Guangzhou, Cina).

Data-aware Processes. WhiSeMech will consider mechanisms that deal with data. It is known that verification and even more synthesis of data-aware processes are in general problematic, since processes generate infinite-state transition systems. However the PI has already shown, within the EU FP7-ICT-257593 ACSI: Artifact-Centric Service Interoperation, that such difficulties can be overcome in notable cases [7, 14, 1, 16]. Since then important advancements in undestanding how to deal with such complexity have been established as well as relationships with formalisms for reasoning about action [50, 6, 18, 17]. On this theme, the PI team will collaborate with Rick Hull (IBM Research, USA), Jianwen Su (UCSB, USA), Diego Calvanese (U. Bolzano, Italy) and Marco Montali (U. Bolzano, Italy), and with Alessio Lomuscio (Imperial College, London, UK).

Verification and Synthesis. WhiSeMech will make use of the mathematical elegant theory of **Reactive Synthesis** [65] developed in formal methods in the last 30 years, which however has not

found diffused practical application because of the **intrinsic difficulties** of certain algorithms and constructions [47]. We aim at **sidestepping these difficulties all-together**, by focusing on non-traditional kinds of specification formalisms. Examples of these are Linear-time Temporal Logic and Linear Dynamic Logic on finite traces, recently proposed by the PI together with Moshe Vardi (Rice U, Huston) [44, 41, 42]. On this theme, the PI team will collaborate with Moshe Vardi (Rice U.) on automata-based verification and synthesis, and Nello Murano (U. Napoli, Italy), Sasha Rubin (U. Napoli, Italy) and Benjamin Aminof (TU Wien, Austria) on game-based verification and synthesis.

Planning. While WhiSeMech will consider symbolic techniques adopted in **synthesis by model checking** [9], it aims at leveraging on the exceptional scalability improvements of current algorithms in **planning in AI**, to devise radically different techniques to effectively tackle reactive synthesis in practice [48]. The PI has been pioneering cross-fertilization of planning and synthesis since [43, 19, 73, 27, 64, 37]. More recently the PI has established tight connection betweens synthesis and generalized forms planning [51, 52, 38, 10] as well as between planning and behavior compositions [70, 39, 28, 15]. On this theme, the PI team will collaborate with Hector Geffner (UPF, Barcelona), Blai Bonet (U. Simon Bolivar, Caracas), Alfonso Gerevini (U. Brescia, Italy) and Malte Helmert (U. Basil). Moreover, in collaboration with Ronen Brafman (Ben-Gurion U.) WhiSeMech will explore temporally-extended goal planing and synthesis in non-Markovian MDPs and reinforcement learning (first ideas in [12]).

Applications. WhiSeMech will ground its scientific results in diverse real **application contexts**, including manufacturing systems (Industry 4.0), smart spaces (IoT), and business processes (BPM) to demonstrate the actual utilization of the scientific achievements within the project. The PI and his group at Sapienza has contributed to all these fields, see e.g., [25, 26, 45]. Moreover the PI has shown to be able to apply advanced science to real-cases in the area Semantic Data Integration where: he has been as one of the main proposers of the Ontology based Data Access paradigm, possibly the most successful approach for Semantic Data Integration [66, 75]; he contributed to W3C recommendation of OWL 2 Web Ontology Language Profiles (https://www.w3.org/TR/owl2-profiles/); and he has founded a Sapienza Start-Up **OBDA Systems** (http://www.obdasystems.com) to commercially exploit Ontology based Data Access in real data integration scenarios.

Project structure. The scientific work in WhiSeMech will be divided into 3 research streams of the duration of the project.

- **Stream 1: Foundations.** This stream will deal with the scientific foundations of white-box self-programming mechanisms.
- **Stream 2: Algorithms and Tools.** This stream will deal with the development of practical algorithms, optimizations and tools for realizing white-box self-programming mechanisms.
- **Stream 3: Applications and Evaluation.** This stream will study evaluate white-box self programming mechanisms in the three business critical application contexts mentioned above.

HIGH RISK, HIGH GAIN

WhiSeMech is a **high risk, high gain project**: if successful, it will result in a radically more useful automated mechanisms than what we have today, namely **white-box self-programming mechanisms**, unleashing full potential of **self-programmability** and removing the main barriers to the uptake of automated mechanisms in real business context, namely *predefined forms of automation*, and difficulties in *formally analyzing their automated behavior in human terms*.

Despite the challenges that WhiSeMech will face, it can take advantage of an emerging convergence between research on **Reasoning About Action**, **Data-aware Processes**,**Verification and Synthesis** and **Planning**, which are the most prominent areas developing methodologies, algorithms and tools related to **white-box self-programming mechanisms**. This convergence has already allowed the PI to show decidability of data-aware processes in spite of them being infinite state due to data [1, 31, 17], and to bringing about new feasibility results, which sidestep some intrinsic difficulties of reactive synthesis algorithms and constructions (e.g., determinization) making synthesis practically feasible in notable cases [44, 41, 42].