DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Relatório Técnico

RT-MAC-2017-03

FLAVIO S. CORREA DA SILVA, PETROS PAPAPANAGIOTOU, DAVID MURRAY-RUST, AND DAVID S. ROBERTSON

SOFT INSTITUTIONS – A PLATFORM TO DESIGN AND IMPLEMENT SOCIO-TECHNICAL SYSTEMS

Fevereiro de 2017

Soft Institutions – a platform to design and implement socio-technical systems

Flavio S. Correa da Silva
LIDET - USP

Petros Papapanagiotou
University of Edinburgh

David Murray-Rust
University of Edinburgh

David S. Robertson
University of Edinburgh

Abstract

In the present article we focus on the notion of *socio-technical systems* and argue that *soft institutions* can be adopted as a computational platform to design, implement and manage such systems. socio-technical systems are characterised as complex networks of heterogeneous interacting peers comprised by humans as well as designed artefacts. The core notion in soft institutions are interaction protocols, which are collections of plans for message passing, state management and action taking which can be used as partial specifications of peer interactions. We revise the notions of both socio-technical systems and soft institutions, and then illustrate how they can be blended together.

1 Introduction

In the present article we focus on the notion of *socio-technical systems* and argue that *soft institutions* can be adopted as a computational platform to design, implement and manage such systems. The concept of socio-technical systems is not new [10]. Recently it has been applied to a variety of complex systems in which the social and the technological dimensions must be considered together [1, 2]. socio-technical systems are open, asynchronous concurrent systems in which some processing units are humans or human centred organisations and others are as machines. Interactions involving heterogeneous peers are a central concept to design, implement and analyse socio-technical systems. In the present article we introduce a platform to characterise interactions, catering for the dynamics and evolution of such systems. This platform can be useful for systems designers, systems engineers and users: (1) Systems designers need tools and languages to specify systems, to extract properties from design (automatically whenever possible), to perform simulations and to extract properties from simulations, to collect field data and to extract properties from field data in similar way as is done with simulations, to feed back the design process and to incrementally refine socio-technical systems. (2) Systems engineers need tools and languages to take blueprints from designers and implement them for actual use, in the form of platforms that mediate interactions among heterogeneous peers in such way that a socio-technical system can be put to work. (3) Users can be the human components that participate in a socio-technical system or external users that expect outputs generated by the system as a whole. Both types of users need appropriate ways to interact with a socio-technical system, as well as means to influence the evolution of the system.

In section 2 we detail a characterisation of socio-technical systems and propose five dimensions which can be used to classify particular systems with respect to design strategies that can be used for them. In section 3 we introduce the concept of *soft institutions*, a corresponding computational platform based on this concept and how it can be used as a platform to design, implement and manage socio-technical systems. In section 4 we discuss an illustrative example of socio-technical system that can be build using soft institutions. Finally, in section 5 we present a brief discussion, conclusions and proposed future work.

2 Socio-technical systems – facets and dimensions

A socio-technical system is an open network of heterogeneous interacting components which can exchange messages and, therefore, coordinate their actions. Some components are engineered and can be programmed to behave according to explicitly determined rules, while other components are human centric and therefore their behaviour can, at best, be incentivised and nudged towards desired patterns.

In [2] we find a framework that characterises the facets of socio-technical systems based on six categories of interacting elements. Our goal in the present work is to introduce a computational platform to support the design, implementation and management of socio-technical systems. For this reason, we identify, for each category of elements in these systems, specific entities that can be characterised and dealt with in the proposed platform: (1) **People** correspond to human agents, whose behaviour can be steered towards desired patterns via appropriate sets of rules, sanctions and rewards which, in turn, must be permanently monitored and refined to ensure their effectiveness; (2) **Technologies** refer to engineered peers as well as communication technologies for peers to interact with each other and with external system users; (3) **Processes/procedures** refer to (a) sets of rules, sanctions, rewards and planning resources and capabilities proposed to humans, (b) programming resources and techniques used in engineered peers, and (c) programming resources and techniques available to build a communications infrastructure for a socio-technical system; (4) **Buildings/infrastructure** are generalised as any set of environmental factors that characterise the (evolving and dynamic) context in which a socio-technical inhabits and operates; (5) **Goals** characterise global, system level goals as well as local, agent level goals, which must be aligned by designers in such way that any action supports and benefits local as well as global goals; and (6) **Culture** characterises (defeasible and dynamic) values and priorities determined for (and by) individuals, groups and the whole system.

Depending on the combination and organisation of components, different design strategies and techniques can be appropriate. For this reason, we identify five dimensions which can characterise different socio-technical systems: (1) **Openness** to admit or dismiss agents; (2) **Coordination** levels among agents; (3) **Heterogeneity** of agents in a system; (4) **Statefulness**; and (5) **Context sensitiveness**.

Different combinations of values of these dimensions require different strategies for design, implementation and management of socio-technical systems. Any proposed platform to deal with socio-technical systems must cater for a wide set of combinations of these values, even if it is not capable of coping satisfactorily with all of these combinations. In the next section we introduce a proposed platform, coined *soft institutions*.

3 Soft institutions

Soft institutions generalise the concept of electronic institutions [3, 4, 9] to provide means for more natural interactions with human peers [5].

Electronic institutions are a powerful framework to build multi-agent systems based on the principle that the global behaviour of a complex system can be managed by the establishment of (1) norms, (2) rewards for agents that abide by these norms and (3) sanctions for those who challenge them. In order for an agent to participate in an electronic institution, it must be prepared to respond to norms, rewards and sanctions, as well as interact with other participating agents.

Technological agents can be designed and built to comply with normative systems and, therefore, participate in electronic institutions. Human agents, however, may feel uncomfortable to need to learn and then to be submissive to third party rules to join into a network of peers. Soft institutions, in contrast with electronic institutions, allow agents to act freely and adjust their behaviour in a minimalist way to be able to join into local interaction protocols. Instead of having a centralised control around the normative system (as is the case with electronic institutions), soft institutions have a decentralised, possibly asynchronous control, centered on agents which choose to interact according to available protocols. This way, the barrier to enter a soft institution is significantly lower for humans, hence an interaction platform based on soft institutions can be more appealing to human agents than one based on electronic institutions, at the cost of only being able to have partial control over design, operation and management of a system based on soft institutions.

Soft institutions are organised in four layers: (1) **The agent controlled layer**: this layer caters for individual capabilities and actions corresponding to each agent. Agents can be human individuals, technological agents, or organisations constituted of other agents; (2) **The communications layer**: this layer comprises the infrastructure and processing power to manage message exchanges between agents. In principle, messaging is peer-to-peer with unique addressing. Additional message control structures can be built using the agent controlled and the communications layer. (3) **The coordination layer:** this layer consists of social norms that constrain and regulate interactions among selected peers. (4) **The environment:** this layer comprises all other phenomena that can influence the behaviour and state of the soft institution.

We assume a language $\mathcal{L}$ used to describe facts and computational expressions. The language consists of three constructs: (1) **Terms:** correspond to constant or atomic expressions of different types; (2) **Variables:** are uniquely identified strings to which different values can be assigned; (3) **Functions:** are collections of mappings from tuples of terms to terms. Value assignments to variables are expressed as substitutions $\sigma = \{x_1 \mapsto c_1, \cdots, x_n \mapsto c_n\}$, which denote that each construct c_i is assigned to the variable x_i. A substitution application function $\hat{\sigma}$ is applied to a construct c as a whole, producing a new construct in which every variable in c that is present in a substitution σ is replaced by the corresponding construct. For example, if $\sigma = \{x_1 \mapsto y, x_2 \mapsto 5\}$ and $c = (x_1 + x_2 + x_3)$, then $\hat{\sigma}c = (y + 5 + x_3)$. Using substitutions we can naturally define unification ($\stackrel{\hat{\sigma}}{=}$). A substitution application $\hat{\sigma}$ unifies two constructs c_1 and c_2 if the application of $\hat{\sigma}$ to both constructs yields the same result, i.e. $c_1 \stackrel{\hat{\sigma}}{=} c_2$ *iff* $\hat{\sigma}c_1 = \hat{\sigma}c_2$.

Each agent maintains a personal knowledge base that comprises its beliefs, opinions, individual goals, actual knowledge, reasoning capabilities, actions etc. It is assumed, as a design principle, that agents do not have access to each others' personal knowledge bases. It is also assumed, however, that each agent participating in a soft institution maintains a part of its knowledge base stored as a collection of $\mathcal{L}$ constructs, which we here name *institutional knowledge base*, and which are updated and consulted using two operators: (1) $\mathbb{A}(c)$: this operator updates a fact c. Depending on specific institutions being designed, an update may correspond to inserting, actual updating or deleting information from the institutional knowledge base; (2) $\mathbb{K}(c, \hat{\sigma})$: this operator consults the institutional knowledge base. Similar to the $\mathbb{A}$ operator, variations on the semantics of the $\mathbb{K}$ operator can be used for different soft institutions. Essentially, $\mathbb{K}(c, \hat{\sigma})$ checks whether the construct c belongs to the institutional knowledge base of an agent; if it does, then it is retrieved from the knowledge base, and the substitution $\hat{\sigma}$ is used to build the construct $\hat{\sigma}c$.

The institutional knowledge base contains a set of ground terms $\mathcal{R} = \{R_1, \cdots, R_m\}$ which represent a set of roles available to the agent. It also contains a set of constructs $\mathcal{PROT}$ using the syntax specified in the following paragraphs, which characterise interaction protocols available to the agent. Roles are the means for an agent to enter a soft institution: an agent can pick a role from $\mathcal{R}$, which becomes the institutional role of the agent and grants the agent the right to engage into interactions using an appropriate protocol available in $\mathcal{PROT}$. Roles can be retrieved and updated using the $\mathbb{A}$ and $\mathbb{K}$ operators.

Messages are passed from agent to agent via the communications layer. To each agent is assigned a unique ID, and messages depend upon roles to be properly treated. A message M has the format $M = \langle R_{send}, gT, R_{rec}, ID_{other}\rangle$: R_{send} is the role that the sending agent must necessarily hold when the message is sent; gT is a ground term which corresponds to the content of the message; R_{rec} is the role that the receiving agent must hold in order for the message to be received; ID_{other} is the ID of the "other" agent: it is the ID of the receiver when a message is being sent and the ID of the sender when a message is being received.

The institutional knowledge base also contains two constructs that represent the state of the agent with respect to the soft institution: (1) *Comm* stores the communications state. It contains the agent ID and two message queues containing incoming and outgoing messages respectively. (2) *Coord* stores the coordination state. It contains the list of roles already held by the agent including the current role as head of the list, the protocol being followed, the stage of execution of the current protocol and the set of variable assignments/substitutions.

Protocols are defined as a variation and extension of the *Lightweight Coordination Calculus (LCC)* [7], in which A *protocol* is a list of *clauses*. A *clause* defines a script to be followed in order for an interaction to take place. Clauses have the format $cl(R, [c_1, \cdots, c_r]) ::= Def$ where $R \in \mathcal{R}$ is a role, $c_1, \cdots, c_r$ are optional parameters and Def is the body of the clause:

$$\begin{array}{rcl} Def & := & \text{Closed} \mid \text{Out} \mid \text{Out} \leftarrow [\text{In}_1, \cdots, \text{In}_s] \mid \\ & & Def \ \textbf{then} \ Def \mid Def \ \textbf{or} \ Def \\ \text{In}_i & := & \text{rec(Msg)} \mid \text{cond}(c) \\ \text{Out} & := & \text{Null} \mid \mathbb{A}(c) \mid \\ & & \text{chRole}(R', [c'_1, \cdots, c'_{r'}]) \end{array}$$

Closed is a final state that concludes an interaction. Out is an output action: (1) Null is an empty action that does nothing. (2) snd(Msg) sends message Msg to another agent. (3) $\text{chRole}(R', [c'_1, \cdots, c'_{r'}])$ changes the role of the agent during the execution of a clause. (4) $\mathbb{A}(c)$ updates the construct c into the institutional knowledge base. $\text{Out} \leftarrow [\text{In}_1, \cdots, \text{In}_s]$ performs a list of input actions and then performs an output action. An input action In_i is: (1) rec(Msg) receives a message Msg from another agent. (2) cond(c) checks whether there is a construct c' in the institutional knowledge base and a substitution $\hat{\sigma}$ such that $\mathbb{K}(c', \hat{\sigma}) = c$. The construct c is a *condition* which can be satisfied if the answer is positive. **then** is a connective that represents sequential and, i.e. it joins two computational steps in sequence. **or** is a connective that represents non-deterministic choice between two computational steps.

Let us consider a socio-technical system built on top of a soft institution. If an agent decides to interact with other agents in the system, it must formalise an institutional knowledge base and then pick from it a protocol and a role. This formalisation can be performed as a services contract which, once accepted by the agent and a registration authority in charge of the system, provides the agent with a baseline institutional knowledge base containing roles and protocols. After a protocol is selected, the agent can address the messages in the clauses that comprise the protocol to specified agents whose IDs it selects, or send messages to arbitrary agents by supplying free variables instead of IDs corresponding to receiving agents, or even broadcast messages to several agents and wait for the best reply in order to choose an agent to complete a protocol. The agent can also select protocols in which service requests can be received and act as a service provider.

Carefully crafted sets of protocols can implement sophisticated patterns of interaction, servicing large and complex socio-technical systems. Interaction protocols work as support services for agents to engage into well regulated and carefully designed interactions, but they are not mandatory and they do not necessarily cover all aspects of all interactions that connect agents participating in the same socio-technical system.

4 An illustrative example

In this section we consider a snippet of a socio-technical system built to support commercial transactions between human agents. This example is a simplified version of an example originally used to illustrate the use of LCC for peer-to-peer knowledge sharing [8]. Let us assume that we have two agents whose IDs are resp. A and B. Agent A wishes to buy an object C and agent B is willing to sell the same object C. Let us also assume that both A and B enter the same soft institution. A acquires an institutional knowledge base which contains an interaction protocol in which the following clause can be found:

$$
\begin{aligned}
&cl(\text{buyer}, [C, Price]) ::= \\
&\quad \text{snd}(\langle \text{buyer}, \text{ask}(C), \text{seller}, Any\rangle) \ \textbf{then} \\
&\quad \text{snd}(\langle \text{buyer}, \text{buy}(C, Price), \text{seller}, Any\rangle) \\
&\qquad \leftarrow [\text{rec}(\langle \text{buyer}, \text{price}(C, Price), \text{seller}, Any\rangle), \\
&\qquad\quad \text{cond}(\text{afford}(Price))] \ \textbf{then} \\
&\quad \mathbb{A}(\text{own}(C)) \\
&\qquad \leftarrow [\text{rec}(\langle \text{buyer}, \text{sold}(C, Price), \text{seller}, Any\rangle)]
\end{aligned}
$$

This clause can be read as follows: A can assume the role of "buyer" of object C at price *Price*, and this role will build a successful interaction if$A$ can send a message to an arbitrary seller asking for object C; **then** If the seller sends a message informing a price *Price* and this price is affordable to A, then it sends a second message to the seller informing that it is willing to buy C and pay *Price*; **then** If the seller sends a message informing that the transaction was successful, then A updates the institutional knowledge based by adding the information that it now owns C.

Similarly, B can acquire an institutional knowledge base which contains an interaction protocol containing the following clause:

$cl(\text{seller}, [C, Price]) ::=$
$\quad \text{snd}(\langle \text{seller}, \text{price}(C, Price), \text{buyer}, Any\rangle)$
$\quad\quad \leftarrow [\text{rec}(\langle \text{seller}, \text{ask}(C), \text{buyer}, Any\rangle),$
$\quad\quad\quad \text{cond}(\text{in-stock}(C, Price))]$ **then**
$\quad \text{snd}(\langle \text{seller}, \text{sold}(C, Price), \text{buyer}, Any\rangle)$
$\quad\quad \leftarrow \text{rec}(\langle \text{seller}, \text{buy}(C, Price), \text{buyer}, Any\rangle)$ **then**
$\quad \mathbb{A}(\text{sold}(C))$

This clause can be read as follows: B can assume the role of "seller" of object C at price $Price$, and this role will build a successful interaction given that If B receives a message from an arbitrary buyer asking for object C and object C is available in stock and can be sold at price $Price$, then B sends a message to the buyer informing availability and price of C; **then** If B receives a message from the buyer confirming the acquisition of C, it sends a message to the buyer informing that the transaction was successful; **then** B completes the protocol by updating the institutional knowledge base by adding the information that C has been sold.

In this particular example, both A and B were not concerned in advance about connecting specifically to each other: A was interested in finding a seller and B was interested in finding a buyer. Prototypical implementations of platforms for soft institutions are under development. A fully functioning platform shall include communication channels that can be used during the execution of interaction protocols, thus enabling that agents A and B can find each other and instantiate the Any variables in their messages with each others' IDs.

5 Conclusion and future work

In the present article we have presented the concept of socio-technical systems based on six facets and five dimensions, and then we have argued that soft institutions can form a computational platform to design, implement and manage socio-technical systems. **People** can easily join into a socio-technical system based on soft institutions – the required effort corresponds to registering into the soft institution (which can have a low barrier to entry by integrating, for example, with existing social networks), downloading and installing a baseline institutional knowledge base, and ensuring that the interactions that are selected to be mediated by the soft institution occur via interaction protocols; **technological agents** can also be designed such that interactions are performed using LCC; **processes/procedures** can be partially formalised using interaction protocols; **environmental factors** can be abstracted as external variables updated by sensors and messages incoming from the environment; **goals** and **culture** are implemented internally to agents. **Openness** can be managed in the registration procedures for humans as well as for technological agents; **coordination** can be achieved by the establishment of joint standards to be followed in every interaction protocol, e.g. related to security verification and structured message passing; **heterogeneity** of agents is indeed one of the strongest points of soft institutions as means to build socio-technical systems, as humans and technological agents are treated equally with respect to message passing and observed behaviour; **statefulness** is managed by bookkeeping institutional knowledge bases; **context sensitiveness** occurs at the agents level, meaning that an individual agent changes behaviour according to the context in which it *believes* to be, thus reproducing what occurs in social systems in general.

Implementations of platforms for soft institutions have already been presented elsewhere [5]. We are expanding that work with formalised semantics that enable the formal verification of interaction protocols with respect to desired properties, leading to increased levels of trust and transparency [6]. Our focus, at the moment, is on applying the ideas presented here in three different domains: (1) safety analysis for complex cyber-physical systems, with particular focus on air traffic control for scenarios in which pilot controlled vehicles must interact with autonomous vehicles[1]; (2) decentralised Local Exchange Trading Schemata (LETS), which are "local community-based mutual aid networks in which people exchange all kinds of goods and services with one another, without the need for money"[23]; and (3) mesoscale phenomena related to crowd management, following a line of work started by Clegg and colleagues [2][4].

Concrete results about these applications shall be presented in future publications.

Acknowledgements

This work has been partially supported by FAPESP-Brazil.

References

[1] G. Baxter and I. Sommerville. Socio-technical systems: From design methods to systems engineering. *Interacting with computers*, 23(1):4–17, 2011.

[2] M. C. Davis, R. Challenger, D. N. W. Jayewardene, and C. W. Clegg. Advancing socio-technical systems thinking: a call for bravery. *Applied Ergonomics*, 45:171–180, 2014.

[3] M. Esteva, J. A. Rodriguez-Aguilar, C. Sierra, P. Garcia, and J. L. Arcos. On the formal specification of electronic institutions. In *Agent mediated electronic commerce*, pages 126–147. Springer, 2001.

[1]This is joint work with Prof. P. W. H. Chung, Loughborough University (UK).

[2]Extracted from `http://www.letslinkuk.net/`

[3]This is joint work with Prof. W. Vasconcelos, University of Aberdeen (UK), Prof. F. Cabitza, University of Milano-Bicocca (Italy) and Dr. T. Sultana, University of Chittagong (Bangladesh).

[4]This is joint work with Prof. W. Vasconcelos, University of Aberdeen (UK) and Prof. S. Bandini, University of Milano-Bicocca (Italy).

[4] M. Esteva and C. Sierra. *Electronic Institutions: from specification to development.* Consell Superior d'Investigacions Científiques, Institut d'Investigació en Intel· ligència Artificial, 2003.

[5] D. Murray-Rust, P. Papapanagiotou, and D. Robertson. Softening electronic institutions to support natural interaction. *Human Computation*, 2(2), 2015.

[6] P. Papapanagiotou, D. Murray-Rust, and D. Robertson. Evolution of the lightweight coordination calculus using formal analysis. *Personal communication*, 2016.

[7] D. Robertson. *Multi-agent coordination as distributed logic programming*, pages 416–430. Proceedings 20th International Conference on Logic Programming – Springer LNCS 3132. 2004.

[8] D. Robertson, A. Barker, and P. Besana. Models of interaction as a grounding for peer to peer knowledge sharing. In *Advances in Web Semantics I*, pages 81–129. Springer, 2008.

[9] C. Sierra, J. A. Rodriguez-Aguilar, P. Noriega, M. Esteva, and J. L. Arcos. Engineering multi-agent systems as electronic institutions. *European Journal for the Informatics Professional*, 4(4):33–39, 2004.

[10] E. Trist. The evolution of socio-technical systems. *Occasional paper*, 2:1981, 1981.

RELATÓRIOS TÉCNICOS

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
Instituto de Matemática e Estatística da USP

A listagem contendo os relatórios técnicos anteriores a 2012 poderá ser consultada ou solicitada à Secretaria do Departamento, pessoalmente, por carta ou e-mail (mac@ime.usp.br).

PAULO H. FLORIANO, LUCIANA ARANTES AND ALFREDO GOLDMAN
CONDIÇÕES DE CONECTIVIDADE DE ALGORITMOS DE EXCLUSÃO MÚTUA EM REDES DINÂMICAS
RT-MAC-2012-01 – fevereiro 2012, 21 pp.

VIVIANE SANTOS AND ALFREDO GOLDMAN
FOSTERING INTER-TEAM KNOWLEDGE SHARING EFFECTIVENESS IN AGILE SOFTWARE DEVELOPMENT
RT-MAC-2012-02 – abril 2012, 42 pp.

CLAUDIA DE O. MELO; VIVIANE A. SANTOS; HUGO CORBUCCI; EDUARDO KATAYAMA; ALFREDO GOLDMAN; FABIO KON.
MÉTODOS ÁGEIS NO BRASIL: ESTADO DA PRÁTICA EM TIMES E ORGANIZAÇÕES
RT-MAC-2012-03 – maio 2012, 13 pp.

MIRTHA LINA FERNÁNDEZ VENERO AND FLÁVIO SOARES CORRÊA DA SILVA
*STUDYING THE BEHAVIOR OF JAMSESSION INTERACTION PROTOCOLS USING SPIN**
RT-MAC-2012-04 – junho 2012 – 27pp.

MIRTHA LINA FERNÁNDEZ VENERO AND FLÁVIO SOARES CORRÊA DA SILVA
*A FORMAL SEMANTICS FOR THE **JAMSESSION** COORDINATION PLATFORM*
RT-MAC-2012-05 – junho 2012 – 27pp.

VIVIANE ALMEIDA DOS SANTOS, ALFREDO GOLDMAN VEL LEJBMAN, DÉBORA VASCONCELOS MARTINS, HERNESTO NÓBREGA BORGES FILHO E MARIELA INÉS CORTÉS.
COMPARTILHAMENTO DE CONHECIMENTO ENTRE EQUIPES ÁGEIS: FATORES INFLUENCIADORES
RT-MAC-2012-06 – agosto 2012 – 14pp.

FLÁVIO SOARES CORRÊA DA SILVA, DAVID S. ROBERTSON AND WAMBERTO VASCONCELOS
EXPERIMENTAL INTERACTION SCIENCE
RT-MAC-2013-01 – fevereiro 2013 - 13 pp.

LEONARDO LEITE, NELSON LAGO, THIAGO FURTADO, CARLOS EDUARDO MOREIRA, DANIEL CORDEIRO, DANIEL BATISTA, MARCO AURÉLIO GEROSA, AND FABIO KON
AN ADAPTIVE ENACTMENT ENGINE FOR COMPLEX SERVICE COMPOSITIONS ON THE CLOUD
RT-MAC-2013-02 – fevereiro 2013 - 23pp.

FLÁVIO S. CORRÊA DA SILVA, WAMBERTO VASCONCELOS AND DAVID. S. ROBERTSON
TOWARDS A COMPUTER AIDED SCIENCE OF INTERACTIONS
RT-MAC-2014-01 – janeiro 2014 - 25pp.

GLAUBER DE BONA AND MARCELO FINGER
NOTES ON MEASURING INCONSISTENCY IN PROBABILISTIC LOGIC
RT-MAC- 2014-02 – novembro 2014 – 44pp.

FABIO KON, DANIEL CUKIER, CLAUDIA MELO,ORIT HAZZAN AND HARRY YUKLEA
A CONCEPTUAL FRAMEWORK FOR SOFTWARE STARTUP ECOSYSTEMS: THE CASE OF ISRAEL
RT-MAC-2015-01 – Junho 2015 – 37pp.

FLAVIO S. CORREA DA SILVA, PAUL W. H. CHUNG, SOCIAL CALCULUS, DAVID S. ROBERTSON, DAVID MURRAY-RUST, WAMBERTO W. VASCONCELOS AND PETROS PAPAPANAGIOTOU
LS3C – THE LIGHTWEIGHT, SITUATED, STATEFUL
RT-MAC-2015-02 – Agosto 2015 – 10pp.

DANIEL CUKIER , FABIO KON AND NORRIS KRUEGER
TOWARDS A SOFTWARE STARTUP ECOSYSTEMS MATURITY MODEL
RT-MAC- 2015 -03 – outubro 2015 - 16pp.

MARCO AURÉLIO GEROSA (USP), ANDRÉ CONSTANTINO DA SILVA (IFSP), APARECIDO NILCEU MARANA (UNESP), CARLOS ALBERTO MAZIERO (UFPR), CECÍLIA MARY FISCHER RUBIRA (UNICAMP), DOUGLAS DYLLON JERONIMO DE MACEDO (IFSC), JUSSARA PIMENTA MATOS (IFSP), LISANDRO ZAMBENEDETTI GRANVILLE (UFRGS), NANDAMUDI LANKALAPALLI VIJAYKUMAR (INPE), PAULO LÍCIO DE GEUS (UNICAMP), RICARDO VARELA CORREA (INPE), ROBERTO DE ALENCAR LOTUFO (UNICAMP), GARABED KENCHIAN (IFSP), THOMAS EDSON FILGUEIRAS FILHO (IFSP)
O QUE É INOVAÇÃO E P&D NA INDÚSTRIA DE SOFTWARE?
RT-MAC- 2016 -01 – janeiro 2016 - 16 pp.

FLAVIO S. CORREA DA SILVA, JOAO MARCOS M. BARGUIL, DAVID S. ROBERTSON, WAMBERTO VASCONCELOS, CLAUDIA J. A. ARAUJO
BLOCKCHAIM MEETS SOCIAL COMPUTERS
RT-MAC-2017-01 – fevereiro 2017 – 17 pp.

FLAVIO S. CORREA DA SILVA, PAUL W. H. CHUNG, MARCELO K. ZUFFO, PETROS PAPAPANAGIOTOU, DAVID S. ROBERTSON, WAMBERTO VASCONCELOS
HAZARD IDENTIFICATION FOR COORDINATED UAVs BASED ON SOFT INSTITUTIONS
RT-MAC-2017-02 – fevereiro 2017 – 19 pp.

FLAVIO S. CORREA DA SILVA, PETROS PAPAPANAGIOTOU, DAVID MURRAY-RUST, DAVID S. ROBERTSON
SOFT INSTITUTIONS – A PLATFORM TO DESIGN AND IMPLEMENT SOCIO-TECHNICAL SYSTEMS
RT-MAC-2017-03 – fevereiro 2017 – 09 pp.

DOAÇÃO

De: Depto. MAC
Dt. Rec.: 04.05.2017