that includes a GAG execution engine, a workspace graphical interface and a

We have developed a domain specific language, embedded in Haskell, (AW/GAG-DSL)

user offers, their enacted instances and references to services offered by other users.

workspace (described by a specific GAG) contains the specification of the services the

workspaces communicating asynchronously by exchanges of services. Each user

data-driven processes. The idea behind the model is to have independent user-

declarative model for the specification of distributed collaborative user-centerd and

The Active Workspaces based on Guarded Attribute Grammars (AW/GAG) is a

receiving messages, etc.

operators, running data storage and retrieval queries against a database to

Semantic rules may also have side effects, for instance, computing values using math

for which there exist no (local) rules are considered as call to external services.

if and when its guard becomes satisfied. Finally, tasks on the right-hand side of a rule

evaluated against the effective input values, an operation that filters applicable rules

constraints on the input parameters of tasks are used to restrict the

applicability of a rule in a given configuration. These contraints are guards that are

it out. Constraints on the input parameters of tasks are used to restrict the

decomposition can exist for the same task, each suitable for a specific way of carrying

between a task and its subtasks, and between sibling tasks. Several such

input parameters of tasks, while the semantics rules are used to enforce data-flow

modelled as rewriting rules using the productions of an attribute grammar. The

inherited and synthesized attributes of the grammar respectively model the input and

tasks with input and output parameters. A task is resolved by breaking it into smaller tasks, resolving these subtasks and combining their results to produce the required output. The breaking down of tasks to smaller tasks is

modelled as rewriting rules using the productions of an attribute grammar. The

inherited and synthesized attributes of the grammar respectively model the input and

input parameters of tasks, while the semantics rules are used to enforce data-flow

between a task and its subtasks, and between sibling tasks. Several such

such decomposition can exist for the same task, each suitable for a specific way of carrying

it out. Constraints on the input parameters of tasks are used to restrict the

applicability of a rule in a given configuration. These contraints are guards that are

evaluated against the effective input values, an operation that filters applicable rules

for each pending tasks. Enacting a GAG model is done lazily, that is, a rule is enabled

The Guarded Attribute Grammar (GAG) model developped in SUMO team is centered

around services which are tasks with input and output parameters. A task is resolved

by breaking it into smaller tasks, resolving these subtasks and combining their results

to produce the required output. The breaking down of tasks to smaller tasks is

modelled as rewriting rules using the productions of an attribute grammar. The

inherited and synthesized attributes of the grammar respectively model the input and

input parameters of tasks, while the semantics rules are used to enforce data-flow

between a task and its subtasks, and between sibling tasks. Several such

such decomposition can exist for the same task, each suitable for a specific way of carrying

it out. Constraints on the input parameters of tasks are used to restrict the

applicability of a rule in a given configuration. These contraints are guards that are

evaluated against the effective input values, an operation that filters applicable rules

for each pending tasks. Enacting a GAG model is done lazily, that is, a rule is enabled

if and when its guard becomes satisfied. Finally, tasks on the right-hand side of a rule

for which there exist no (local) rules are considered as call to external services.

Semantic rules may also have side effects, for instance, computing values using math

operators, running data storage and retrieval queries against a database to

respectively save synthesized values and populate inherited attributes, sending and

receiving messages, etc.

The Active Workspaces based on Guarded Attribute Grammars (AW/GAG) is a

declarative model for the specification of distributed collaborative user-centered and
data-driven processes. The idea behind the model is to have independent user-

workspaces communicating asynchronously by exchanges of services. Each user

workspace (described by a specific GAG) contains the specification of the services the

user offers, their enacted instances and references to services offered by other users.

We have developed a domain specific language, embedded in Haskell, (AW/GAG-DSL)

that includes a GAG execution engine, a workspace graphical interface and a

About the research centre or Inria department

The Inria Rennes - Bretagne Atlantique Centre is one of Inria's eight centres and has

more than thirty research teams. The Inria Center is a major and recognized player in

the field of digital sciences. It is at the heart of a rich R&D and innovation ecosystem:

highly innovative PMEs, large industrial groups, competitiveness clusters, research and

higher education players, laboratories of excellence, technological research

institute, etc.

Team presentation :

The SUMO team proposes to combine formal methods approaches with concurrency

theory, in order to address the modeling, analysis and management of large
distributed or modular systems exhibiting quantitative aspects. Large distributed

softwares and systems are indeed calling for quantitative models involving time,

probabilities, costs, and combinations of them. As many problems in this setting

become untractable or even undecidable, we are interested in the design of efficient

approximation techniques, for example borrowed from electrical engineering

approaches to the management of large stochastic systems. A strong point of SUMO

is to gather skills from formal methods, discrete event systems, concurrency theory,

and electrical engineering. Several application fields are covered: telecommunication

networks management, modeling and verification of web services, control issues in

large data centers, plus more opportunistic applications in the field of embedded

systems or biological pathways.

Context

The Guarded Attribute Grammar (GAG) model developped in SUMO team is centered

around services which are tasks with input and output parameters. A task is resolved

by breaking it into smaller tasks, resolving these subtasks and combining their results to

produce the required output. The breaking down of tasks to smaller tasks is

modelled as rewriting rules using the productions of an attribute grammar. The

inherited and synthesized attributes of the grammar respectively model the input and

input parameters of tasks, while the semantics rules are used to enforce data-flow

between a task and its subtasks, and between sibling tasks. Several such

such decomposition can exist for the same task, each suitable for a specific way of carrying

it out. Constraints on the input parameters of tasks are used to restrict the

applicability of a rule in a given configuration. These contraints are guards that are

evaluated against the effective input values, an operation that filters applicable rules

for each pending tasks. Enacting a GAG model is done lazily, that is, a rule is enabled

if and when its guard becomes satisfied. Finally, tasks on the right-hand side of a rule

for which there exist no (local) rules are considered as call to external services.

Semantic rules may also have side effects, for instance, computing values using math

operators, running data storage and retrieval queries against a database to

respectively save synthesized values and populate inherited attributes, sending and

receiving messages, etc.

The Active Workspaces based on Guarded Attribute Grammars (AW/GAG) is a

declarative model for the specification of distributed collaborative user-centered and
data-driven processes. The idea behind the model is to have independent user-

workspaces communicating asynchronously by exchanges of services. Each user

workspace (described by a specific GAG) contains the specification of the services the

user offers, their enacted instances and references to services offered by other users.

We have developed a domain specific language, embedded in Haskell, (AW/GAG-DSL)

that includes a GAG execution engine, a workspace graphical interface and a

2018-00626 - post-doctoral fellowship on a
domain specific language for crowd-sourcing
systems specified by Guarded Attribute
Grammars (SUMO)

Level of qualifications required :PhD or equivalent

Fonction : Post-Doctoral Research Visit

General Information

• Theme/Domain : Proofs and Verification
• Town/city : Rennes
• Inria Center : CRI Rennes - Bretagne Atlantique
• Starting date : 2018-11-01
• Duration of contract : 1 year, 2 months
• Deadline to apply : 2018-05-31

Contacts

• Inria Team : SUMO
• Recruiter : Badouel Eric / eric.badouel@inria.fr

Conditions for application

Thank you to send us these documents by applying online:

- updated CV
- cover letter
- letters of recommendation eventually
- degree transcripts

More informations : eric.badouel@inria.fr
middleware in charge of communication between workspaces, user administration and task assignment.

We’re working on this with

- The startup OpenAgora (https://www.open-agora.com/fr) on collective decision-making systems;
- DRUID team, in the frame of ANR Headwork, on workflow models for crowd-sourcing applications;
- Milan University and the JRC in Ispra on the formalisation of debates (in the broad sense: parliamentary, citizens, experts, etc.) and the management of related documents;
- CESPA (Centre d’épidémiologie et de santé publique des armées, Marseille) on health crisis management systems.

Travel related to this project will be covered within the limits of the scale in force.

Assignment

With the help of Eric Badouel, the recruited person will develop the AW/GAG-DSL for crowd-sourcing applications and conduct experiments with our partners. The study will focus on the middleware part of the AW/GAG-DSL.

From a formal perspective, we are seeking mechanisms to produce knowledge from information transmitted by crowd participants given that this information is imprecise, uncertain and most often than not contradictory (presence of conflicts). Emphasis will be placed on point of view extraction (using clustering techniques), amendment management and consensus building mechanisms.

These mechanisms will be implemented in the AW/GAG-DSL and then used to specify and experiment collaborative systems in the context of two case studies. The first, which follows Robert Nsaibiri's thesis on the development of an early detection system for epidemics (in partnership with CESPA), will focus on health crisis management systems. The second, in collaboration with the University of Milan and the JRC of Ispra, will focus on the formalization of debates and the management of associated documents.

Main activities

Main activities of the recruited person are thus:

- Design mechanisms for merging data produced by a pool of users.
- Design clustering algorithms to identify the various points of view and related conflicts that occur in a debate as well as amendment and negotiation mechanisms in consensus building.
- Contribute to the development of the AW/GAG domain specific language to incorporate these mechanisms in order to make it usable in a crowd-sourcing context.
- Develop case studies in collaboration with our partners
- Write reports and submit research articles.

Skills

The proposed post-doctoral fellowship requires a taste and competencies both for formal models (attribute grammars, Dempster-Shaffer Theory of evidence ...), algorithmics and programmation (in Haskell), and experimentation on real world issues.

Benefits package

- Subsidised catering service
- Partially-reimbursed public transport
- Social security
- Sports facilities

Remuneration

Gross salary : 2653 euros