Ecosystems Enabling Adaptive Composition of Intelligent Services

César A. Marín

Service Systems Research Group, MBS.
The University of Manchester.
Manchester M15 6PB, United Kingdom
Email: cesar.marin@manchester.ac.uk

José Barbosa and Paulo Leitão

Polytechnic Institute of Bragança.
5301-857 Bragança,
Portugal
Email: {jbarbosa,pleitao}@ipb.pt

***Abstract*—Intelligent services are roughly defined as pieces of software with the capabilities of problem-solving and autonomous composition of solutions, e.g., composition of manufacturing processes. They are heterogeneous and distributed by design, however in industrial settings they are constrained to local interactions with limited room for adaptation due to the need to lower the interoperability barrier. In this paper, we present an approach where collections of intelligent services are treated as ecosystems, using food chains, environments and migration to enable adaptive compositions that generate solutions with a higher service value chain. We present a set of experiments demonstrating how a distributed ecosystem achieves compositions of solutions with higher service value chains while balancing the load and diversity of intelligent services across the ecosystem via self-organisation. This supports the claim that implementations of intelligent service based systems (ISBS) as ecosystems could bring substantial benefits to industrial applications.**

Keywords–Ecosystem; Intelligent services; Adaptive composition.

I. Introduction

Recent developments in industrial applications of agent [1] and service [2] technologies have seen the combination and redefinition of the term *intelligent services*. Intelligent services have been defined as "independent pieces of software that are expected to provide a particular result, either produced by the intelligent service itself or by requesting support from other intelligent services" [3]. They are used to compose solutions to problems in an autonomous manner, e.g., composition of manufacturing processes [3] [4]. By definition, intelligent services accommodate the existence of heterogeneous implementations such as software agents, web services, or any of their combinations. This creates an interoperability barrier that is lowered by the utilisation of a central Enterprise Service Bus (ESB) for multi-protocol communication [5].

The result is an ISBS distributed by design, but centralised by implementation due to the ESB because all intelligent services have to connect to it for communication. Consequently, the intelligent services are constrained to local interactions with limited room for adaptation. This calls for an approach that unlocks the potential of intelligent services by reaching outside their centralised implementation while benefiting from an environment where the interoperability barrier has been lowered.

The contribution of this paper is an approach where ISBS are treated as ecosystems, where food chains, environments and migration trigger an adaptive composition that obtains solutions that benefit from higher a service value chains across ecosystems. Our experiments demonstrate that the service value chain of composed solutions increases while the system regulates its diversity. This sustains the claim that industrial applications of intelligent services could benefit from using our ecosystem approach to achieve more stable, balanced, efficient and adaptive distributed systems for their own support.

The remaining of this paper is structured as follows: Section II provides more details of ISBS. Section III introduces the approach of using ecosystems as a way to model intelligent service compositions. Section IV explains the experiments carried out and analyses the achieved results. Section V discuss related work done in the area. Finally, Section VI round up the paper with a conclusion.

II. Intelligent Service Based Systems

Cyber-Physical Systems are seen as the way to support the fourth industrial revolution where all devices that act on the industry will be connected. These systems promote the decentralisation of the control over a set of distributed entities where, through their cooperation, the global system behaviour is achieved. One possibility to develop the interface layer of these systems is by the adoption of functionality exposure to others using service oriented architecture. To this extend, the current use of services is not sufficient, since in general they do not exhibit intelligence which could allow them to make on-the-fly adaptations as the surrounding constraints change. In this way a different view of services is needed where behaviour is embedded.

We take the definition of *intelligent services* from [3] which defines them as "independent pieces of software that are expected to provide a particular result, either produced by the intelligent service itself or by requesting support from other intelligent services". This implies that intelligent services possess the capabilities of problem-solving and autonomous composition of solutions, e.g., process composition in manufacturing. Particularly, intelligent services are not bound by any specific technology nor platform, but rather can be seen as autonomous entities that expose their internal functionalities as services. In practice, they are the combination of two worlds: agents providing autonomy and intelligence, and services offering ease of aggregation [6].

Autonomous agents can be used to provide the services with the "intelligent" part allowing the reasoning and adaptation behind the service encapsulation. To this extent, basically three types of combinations can be envisioned [7]. The first approach is to use gateways for semantic translation from the agent world to the service world; Another approach is to encapsulate single agents as services, thus having a direct access to other services; The last approach is to use service-oriented agents that not only share services as their major form of communication, but also complement their own goals with external provided services.

By definition, intelligent services do not rely on any specific agent-service combination, but rather allow heterogeneous implementation. Therefore, they must rely on a common information exchange platform to lower any interoperability barrier [5]. This can be achieved by means of an ESB, which provides the basic functionalities for communications exchange such as routing and conflict resolution [8], see Figure 1. Transposing this topology into a real world situation, cases are found where a company is composed by multi-site delegations or factories, each one having its own ESB, and thus having the need to share information and services being offered in other companies' ESBs. Different cases can also be found in situations where a set of heterogeneous companies create a working cluster where information and services need also to be shared.

One of the issues found in implementations of intelligent services is the lack of mechanisms that enable the dynamic replacement of intelligent services in order to better accommodate the structural adaptation around the ESB. This urges the need to create a network of cooperative ISBS in which the offered services of one ISBS migrates into another one according to where they are most needed. This structural change at the ESB level can be achieved by drawing a parallel to natural ecosystems, thus enabling the creation of an ecosystem of ISBS where services migrate from one ISBS to another in their search for better compositions.

III. Ecosystems for Intelligent Services

Natural ecosystems are typical examples of adaptive and self-organising systems where local interactions enable the emergence of complex behaviours [9]. Current implementations of ISBS, e.g., [4], exhibit local interactions which limit their capability for adaptation, a restriction that can be exploited by drawing the parallel between natural ecosystems and ISBS. This enables adaptation of intelligent service compositions across a set of instances of ISBS. Our approach considers three elements of a natural ecosystem that in previous studies have proven successful [9]: *food chains*, *environment* and *migration*. These are explained in the following subsections.

A. Intelligent Service Composition as Food Chains

A food chain is the collection of species in which energy and resources flow from one species to another [10]. Yet species may not only consume from one single species, they may belong to overlapping food chains. Thus the participation of a species in a food chain varies according to the species participation in another food chain, creating shifting networks of energy and resource flows [11]. Consequently, food chains are dynamic in such a way that they are created, changed, replaced, dissolved or re-created continually over time. In terms of intelligent services, a food chain is the equivalent of a full composition of intelligent services where a service value chain is generated by aggregating the individual service values from the first intelligent service in the chain up to the last one.

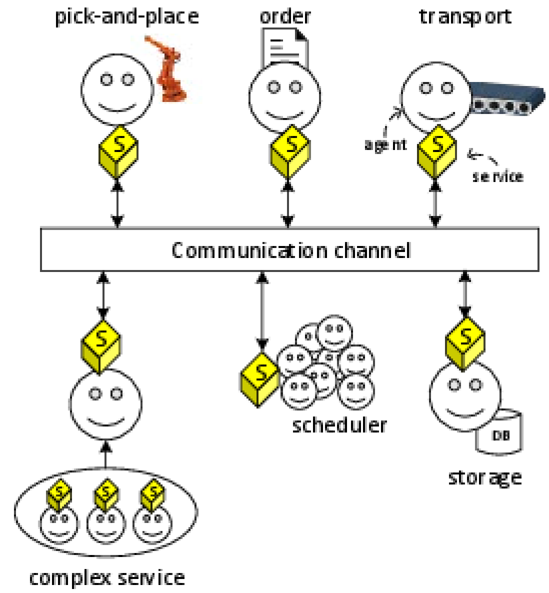

Figure 1. Intelligent services connected to an ESB.

We see each individual in an ecosystem as a member of a specific species which produces a resource of one type and consumes a resource of another type. In terms of ISBS, an individual is the intelligent service itself and the resource is the functionality being offered and consumed by intelligent services. Now let us define the basics for intelligent service composition under the ecosystem approach. For the sake of clarity of and focus on the ecosystem approach, we do not consider any semantic matching or similar. We simply assume that such approaches can be incorporated, e.g., [12].

A link in a composition is formed when the produced resource of an intelligent service is consumed by another intelligent service in such a way that both intelligent services perceive a benefit in maintaining this producer-consumer relationship over others. This is represented as

$$l = i \odot j \quad (1)$$

where i and j are intelligent services and j consumes a resource i produces. The symbol $\odot$ is used to denote that the link formed by two intelligent services represents a resource flow from a producer to a consumer. The operator $\odot$ is not commutative.

The value of a link between the intelligent services i and j is defined as a function in the following way:

$$\nu(l) = \nu(i) + \nu(j) \quad (2)$$

where ν is a function that returns the value of the functionality offered by an intelligent service or a composition of them. For the purpose of this paper such a value is a number used to compare and calculate the value of a composition, the higher the value the better.

A path is a succession of links starting from an intelligent service i to a final intelligent service j where the path is the minimum set of links required to connect them; thus

$$\pi^n(j) = l_0 \odot l_1 \odot \ldots \odot l_{n-1} \quad (3)$$

where $n > 0$ and l_0 begins with intelligent service i and l_{n-1} ends with intelligent service j; that is, $l_0 = i \odot \ldots$ and $l_{n-1} = \ldots \odot j$. It is possible to write $i\,\pi^n j$ to indicate that there exists a path from i to j consisting of n links. Notice that $i\,\pi^1 j \equiv l_0 \equiv i \odot j$.

The notion of *predecessor* and *successor* is used to determine a relative position in a path. Let i and j be intelligent services, it is said that "i is a predecessor of j" or "j is a successor of i" if $i\,\pi^n j$ is a valid path.

The value of a path is calculated by recursively adding up the value of the constituting links from the intelligent service i to j in the following way:

$$\nu\left(\pi^n(j)\right) = \nu\left(\pi^{n-1}(k)\right) + \nu(j) \tag{4}$$

where $n > 0$ and k is the immediate predecessor of j in the path.

A composition of intelligent services, where the final solution is given by intelligent service j, is the collection of paths connecting intelligent services to the final provider j. This is expressed in the following way:

$$\Pi^n(j) = \{ i\,\pi^n j\ ,\ \forall\, i \} \tag{5}$$

where $n > 0$. It possible to write $i\,\Pi^n j$ to denote a composition where i is at the beginning of a constituting path of the composition and j is the final intelligent service providing the final solution. In the simplest case, $\Pi^1(j) \equiv \pi^1(j) \equiv i \odot j$.

Finally, *the service value chain* of the composition $\Pi^n(j)$ is then calculated by adding the value of all the paths where j is the common successor. This is represented in the following way:

$$\nu\left(\Pi^n(j)\right) = \sum^{\Pi^n(j)} \nu\left(\pi^n(j)\right) \tag{6}$$

where $n > 0$. The service value chain is created as resource flow from a single intelligent service all the way through to the final producer of a composite service. The service value chain is used within ISBS to enable the evaluation of compositions, which in turn allow intelligent services to make decisions about who they interact with in order to increase the value of their produced resource, i.e., their offered functionality.

B. Distributed Environment of Migrating Intelligent Services

The environment is an essential element in a natural ecosystem. Without it, species would struggle to survive since it enables them to search for resources to consume. In the context of ISBS, the ESB is the equivalent to the environment. Without it, intelligent services would struggle to interact due to their heterogeneous implementations (see Section II). In natural ecosystems, migration allows species to effectively switch from one environment to another according to the abundance of resources they need. In the context of the ISBS, migrating from one ESB to another would allow intelligent services to relocate to a different environment according to the abundance of resources provided by the intelligent services there.

Our approach considers the environment, cf. [9], as a virtual surface where intelligent services behave as individuals, wandering across and encountering others in order to interact; it mediates interactions and allows intelligent services to forage for resources of their interest. Nevertheless, they keep on exploring the environment for better resources while returning to areas where they have had favourable interactions in the past. The result of this is a dynamic setting where all intelligent services are in motion interacting with others crossing their path.

We see a distributed environment as a collection of interconnected environments forming a network where each of the environments has its own set of intelligent services inhabiting it. Intelligent services then can move from one environment to another in a similar way to species in a natural ecosystem migrate from one environment to another. Since the intelligent services' interactions are local to the environment they occupy, the migration is managed by the environments (i.e., the ESB). Therefore, each environment considers two criteria for enabling migration: 1) the selection of the intelligent service to migrate, and 2) the destination environment.

1) Selection of intelligent service to migrate. We consider two conditions for migration for each type of intelligent service: a) *resource value*, migrate the intelligent service that individually contributes less to any service value chain, and b) *past interactions*, migrate the intelligent service that has interacted less in a certain period of time.

The first condition focuses on moving the intelligent service that, because of its low service value, is likely to be disfavoured as preferred producer, as it is probable for not being selected to be interacted with. By making it migrate, the remaining intelligent services do not waste time in interactions that will not be profitable thus increasing the overall chance of having a greater service value chain.

The second condition focuses on past interactions. We consider each intelligent service possessing a rolling memory with which they only remember the last n interactions. Therefore, if an interaction with a preferred producer does not repeat before fading away, the preferred producer will be forgotten. This characteristic motivates intelligent services to keep on interacting with their peers as a way to remember preferred interactions while keeping exploring for new and possibly better resources. Consequently, the second condition for migration focuses on that intelligent service that has interacted less in a certain period of time since its exploration has resulted less fruitful. With migration, such intelligent service is given the opportunity to explore in a different environment and thus increasing the chance of being preferred somewhere else.

In both migration conditions the main goal is to allow intelligent services to move to an environment where they can be more useful. Both migration conditions alter the composition of intelligent services and thus the service value chain across the distributed environment. Therefore, the *stability of the compositions* and the *service value chains* are analysed in Section IV.

2) Selection of destination environment for migration. The main concern here is to balance the overload between the environments that constitute the ecosystem, which is based on the current load of intelligent services per produced resource type. The environments are able to calculate their current load and share it with each other, comparing the frequencies of intelligent services local interactions between pairs of environments. The environment pair with the highest difference between frequencies will trigger the migration of an intelligent service between each other, migrating from the high loaded

 ISBN: 978-1-61208-391-9

environment to the less loaded.

```
threshold, migrationCondition
for all IntelServiceTypes do
    maxInteraction ← max(IntelServType)
    minInteraction ← min(IntelServType)
    if maxInteraction − minInteraction > threshold then
        if migrationCondition == resourceValue then
            IntelServToMigrate ← min(lowestResourceValue)
        end if
        if migrationCondition == pastInteractions then
            IntelServToMigrate ← min(lowestInteraction)
        end if
        migrate(IntelServToMigrate)
    end if
end for
```

Figure 2. Migration decision mechanism.

As a consequence of migration, new service compositions emerge in each environment due to the presence of new intelligent services. Likewise, existing compositions are forced to adapt to the new circumstances. Furthermore, the diversity of intelligent services in each environment is affected. In biology, it is well known that the diversity of an ecosystem is a principal ingredient for adaptation and resilience. Analyses of diversity, from the biology point of view, have been carried out in other works to estimate the resilience of information systems [13] [14]. Therefore, the *diversity* of the environments will be measured to estimate the evenness of intelligent services across the distributed environment. See Section IV for more details.

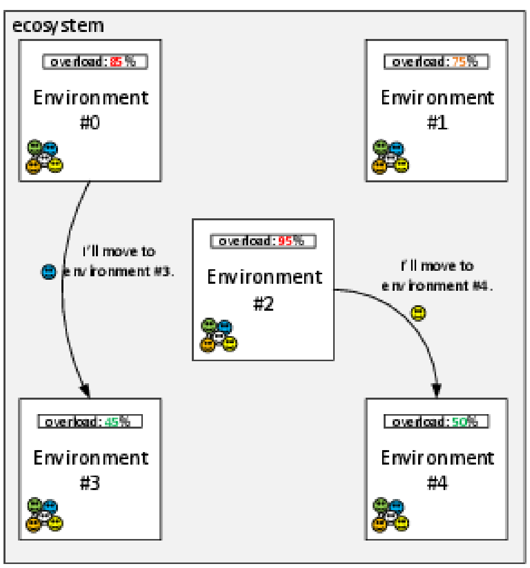

Figure 3. Ecosystem for intelligent services.

3) Combination of migration criteria. The algorithm of the described criteria is shown in Figure 2. It guarantees that at least one intelligent service producing each resource type remains in each environment. This is done to allow at least one full composition to emerge in each environment. Finally, Figure 3 depicts how the distributed ecosystem would look like using five environments. As an example, the interaction load annotated in the figure is used to indicate the environment destination for migration. In the example, it is possible to observe that environment #0 and #2 have high load levels as opposing to environments #3 and #4. Therefore based on the migration criteria, an intelligent service can migrate from a highly loaded environment to one less loaded, trying to balance the environment load while offering the intelligent services the opportunity to maximise their resource value. This triggers both the emergence of new compositions and the adaptation of existing ones.

By having an intelligent service selection criteria and a migration environment selection, the overall ecosystem can remain stable, avoiding disparity of overloaded and underused environments. Despite this environment overload distribution, the generated service value chain can also experience an increase, since intelligent services now have mechanisms to keep interacting with preferred services while constantly seeking for new opportunities to evolve and adapt. In such way, the previously given definition for intelligent services can now be extended by stating that those now have capabilities to switch between ”working” environments aiming to increase their usefulness.

IV. Experiments and Analysis of Results

In order to analyse the advantage of introducing the ecosystem elements of 1) food chains, 2) environment and 3) migration, we use a set of metrics to measure 1) the variation of the service value chains, 2) the stability of compositions, and 3) the diversity of intelligent services across the distributed environment. For this purpose we use the following metrics:

Service value chain. It shows the statistical distribution of the service value chain collected by all the intelligent services acting as top consumers of the composition. The higher and more stable the better.

Stability. It calculates the accumulated number of times the service value chain changes from one simulation step to the next one. The lower and more stable the better.

Diversity. It uses the normalised Shannon Index to measure the level of diversity at a given time in one environment as in [13]. The median of all the normalised Shannon Indexes is then calculated to estimate the diversity across the distributed environment. This is expressed as follows:

$$H' = \frac{-\sum_{i=1}^{S} p_i \; ln(p_i)}{ln(S)} \tag{7}$$

where S is the total number of species (i.e., intelligent service types), p_i is the proportion of individuals of species i (i.e., intelligent services of type i) in the population. The higher the value, the more evenly the species are represented in the environment. The lower the value, the less even is the representation. A normalised diversity of 1 indicates all species have exactly the same number of representatives in that environment. A normalised diversity of 0 indicates all intelligent services in that environment belong to the same type. Therefore, the higher and more stable the value is the better is considered.

A. Experimental Setup

For the experimental test-bed described in this work, Netlogo platform was selected since it aggregates in an overall

 ISBN: 978-1-61208-391-9

manner a good combination of GUIs, ease of programming and an extensive documentation support [15]. We ran two experiments, one per migration condition. **Experiment 1: resource value. Experiment 2: past interactions**. For each experiment, we ran 30 simulations which then were used to calculate the median across the same simulation step. This way, we can appreciate the central behavioural pattern with statistical significance. The median is used as the central representation point because is does not assume any distribution in the sample data.

Each simulation consists of the following consecutive stages:

- Running five independent ecosystems for 1,000 simulation steps, for letting the ecosystems converge to a service value chain. This number of simulation steps was empirically determined to be more than sufficient to detect any convergence in the individual environments.
- Enabling migration between the five environments, rendering a distributed environment (cf. a fully connected network), for 2,000 simulation steps. The number of simulation steps for this stage was empirically determined as well in order to allow the simulations to run for a longer period and be able to notice any trend.

B. *Migration Conditions Increase the Service Value Chain*

Figures 4 and 5 show the service value chain value of experiment 1 and 2 respectively. It can be immediate noticed in both cases that the convergence to a (local) optimum is reached during the first stage. Even when there is no connection between environments, these are in continual internal dynamism because intelligent services interact with whoever they encounter in their environment, keep looking for preferred producers, forget useful interactions, and interact again. Also, notice that the service value chain stabilises even under such dynamism because negative changes are absorbed by these interactions, cf. [4].

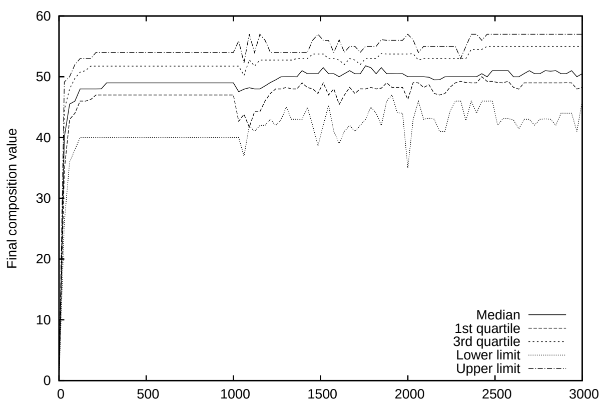

Figure 4. Service value chain value with resource value as migration condition.

The moment migration is enabled, the service value chain immediately varies and the median tends to take slightly higher values. However, when using the resource value as the migration condition, the distribution of the service value chain (Figure 4) tends to be slightly more compact than when

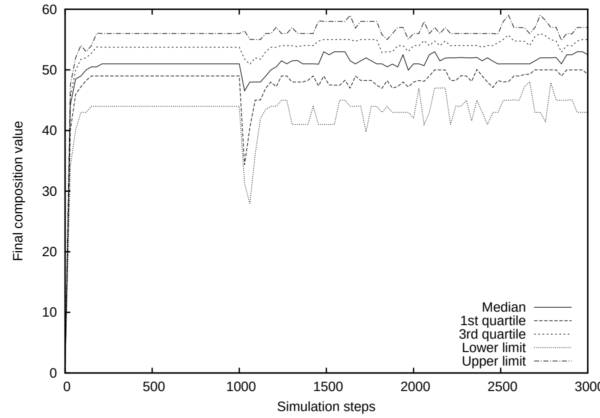

Figure 5. Service value chain with past interactions as migration condition.

past interactions is used (Figure 5). Additionally, migration with past interactions (Figure 5) shows a sharp drop in the service value chain right after migration is enabled. This occurs because in this mode migration does not care about resources, whereas in the other case, intelligent services still try to optimise the resource they produce. Nonetheless, the median shows a slight improvement in the service value chain when migration is enabled.

C. *Past Interactions Render the Ecosystem More Stable*

Figures 6 and 7 depict the stability of the service value chain of experiment 1 and 2, respectively. In both cases, the stability during the first 1000 simulation steps stays under 15 number of changes, this means that environment tend to find and keep an optimum service value chain. However, the moment migration is enabled using resource value, as shown in Figure 6, the number of changes tends to increase and its distribution to widen over time. This is because the motivation of the intelligent services to move is their resource value only, thus making composition more volatile.

In the case of migration with the past interactions condition, as depicted in Figure 7, the number of changes tends to increase as well, but its distribution widens considerably less than its counterpart. This is because the intelligent services that migrate are the intelligent services that interact less in an environment. Consequently, their own resource value is not that essential towards the final composition in that environment. Therefore, if they migrate the remaining intelligent services hardly notice the change.

D. *Past Interactions Maintain Ecosystem Diversity*

Figures 8 and 9 present the diversity across the environments. During the first stage of simulation, both cases show a normalised diversity of 1, meaning that all intelligent service types have exactly the same number of individuals in each environment. This value is expected since no migration has been enabled yet. The moment migration is enabled, a drop in diversity immediately occurs in both cases. This is because initially there is no order regarding where the intelligent services are better required. As the simulations progress and the system starts to adapt, the levels of diversity increase because the environments acquire patterns in terms of proportions of intelligent services per type.

However, when resource value is used as the migration condition (Figure 8), the distribution of diversity widens over

 ISBN: 978-1-61208-391-9

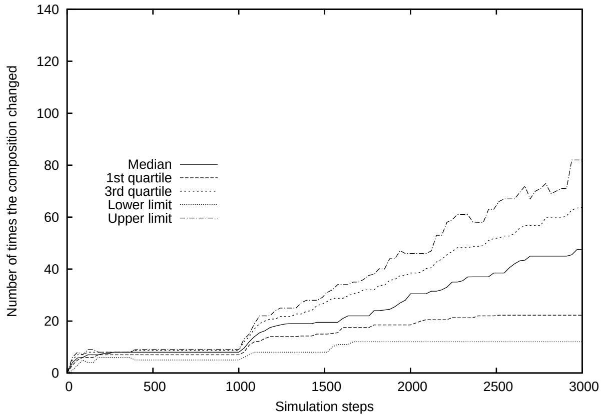

Figure 6. Stability of the service value chain with resource value as migration condition.

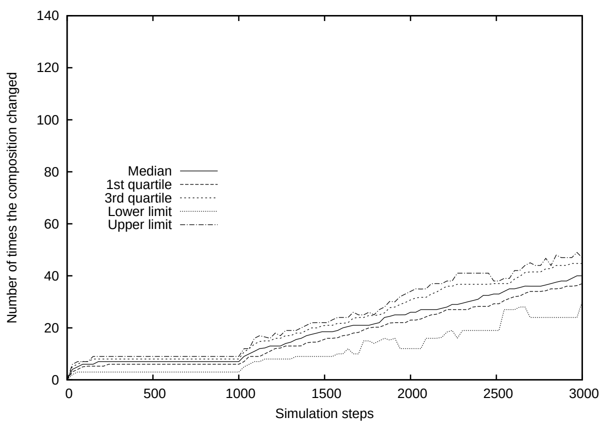

Figure 7. Stability of the service value chain with past interactions as migration condition.

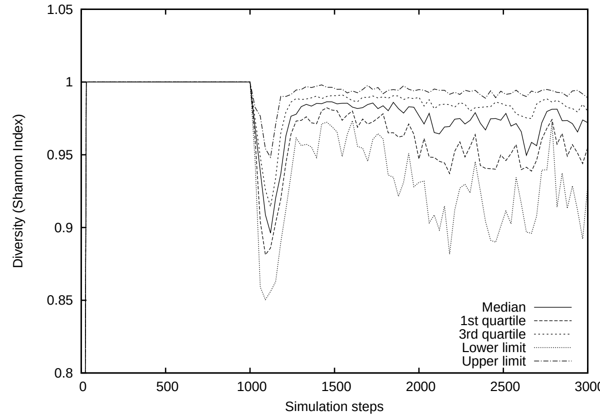

Figure 8. Ecosystem diversity across the environments with resource value as migration decision rule

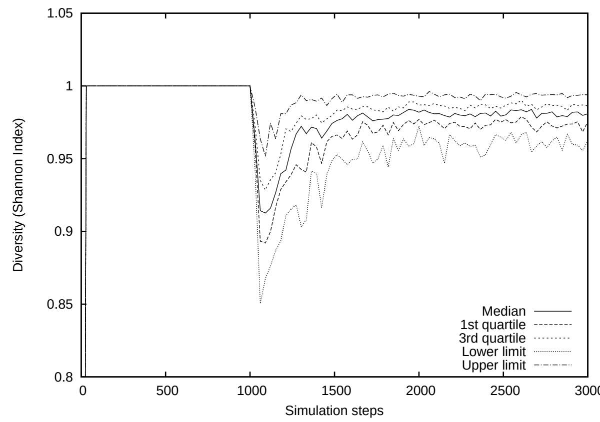

Figure 9. Ecosystem diversity across the environments with oast interactions as migration decision rule

time. This occurs due to the fact that an intelligent service with a low resource value will have a tendency to migrate more often than others, thus varying the diversity of environments and degrading the overall system. On the other hand, when past interactions is used as the migration condition (Figure 9), the distribution of diversity tends to stabilise above 0.95. This is due to the migration condition considering those intelligent services that are not that needed in an environment because they interact less. That is, the structure of the emerging compositions does not require those intelligent services. As a consequence, environment diversity is high and stable.

In summary, a distributed ecosystem environment enabling migration using past interactions as the migration condition brings the following benefits:

- **Improvement of service value chain.** Even when more dynamism is allowed due to environment distribution, the service value chain is increased.
- **Low variation of service value chains.** This renders this migration condition as an enabler of system robustness.
- **High diversity (evenness) of intelligent services across the environments.** This renders this migration condition as a good load balancer thus minimising system degrading.

V. Related Work

Service composition has seen many approaches proposing methodologies and techniques to compose atomic services into more complex ones, fulfilling the users' demand [16] and the ever growing complexity of services. Adaptive and self-organising techniques are emerging as valid approaches to achieve service composition. For example, in [17] a self-organising technique is used to combine service composition with the discovery process into one step. They use an agent based approach aided with contract net protocol to achieve service composition. Then Cloud participants and resources are mapped as agents which negotiate in order to compose the provided services. Additionally, a self-organising multi-agent system approach is used in [7], where agents seeking to be processed are able to search and dynamically organise themselves by composing the offered services, e.g., routing or processing, following an ant based self-organising mechanism. Moreover, [4] presents experiments using ecosystem in the context of manufacturing, where compositions of tasks to realise workflow processes are tested. Regardless of the focus on adaptation and/or self-organisation none of these approaches considers the heterogeneity aspect of intelligent services as our approach does.

In terms of experiments on ecosystems and technologies similar to services, DBE [18] is a platform for supporting business ecosystems. It considers a population of services which a genetic algorithm tries to find the optimal composition with. Evolved service populations live in networked habitats in such a way that successful services tend to cluster where they are most required for compositions. Regardless of the ecosystem inspiration, this approach does not target heterogeneity of services nor adaptation and only focuses on composition optimisation. In another work, [19] presents a comparison of approaches on optimisation of service ecosystems in Cloud environments. The comparison is made using a process template and varying number of services which the approaches have to optimise. Although that work uses ecosystems as part of its context and uses two approaches for services, these approaches are never combined to encompass heterogeneity, a key aspect of intelligent services. In addition, [20] describes an ecosystem inspired architecture for supporting dynamic scenarios such as service ecosystems. The architecture considers entities such as flora and consumers, and tuple species representing niches by which such entities interact. The entities have needs and a "happiness" level they try to optimise by fulfilling their needs. Experiments show a balanced state of "happiness" levels across the predefined niches. However, these experiments only show their capacities of self-organisation and convergence to a solution. In contrast, our experiments not only cover the convergence to a solution, but also how migration keeps a balanced and diverse system without negatively affecting the quality of the solution.

VI. CONCLUSION

The contribution of this paper is an approach where ISBS are treated as ecosystem environments, which form a distributed ecosystem environment when they are interconnected to each other. Three elements of ecosystems are incorporated in our approach: food chains, environment, and migration. An implementation and subsequent experiments demonstrate the advantage of the approach for ISBS.

Our results demonstrate that implementing ISBS as ecosystems using migration of intelligent services based on past interactions, could bring a benefit to the system because a higher service value chain is obtained, which is desirable, when more dynamism is added to ISBS. Even when more dynamism is enabled, both a low variation (i.e., high stability) of composition of solutions and a high diversity of intelligent services across the ISBS are maintained, which minimise system degrading. Our results sustain the claim that ISBS implementing the ecosystem approach presented here can unlock the potential of intelligent services by enabling the adaptation of existing compositions while remaining a robust system. Future work along this line consists of implementing these features in an industrial setting, e.g., manufacturing.

REFERENCES

[1] M. Wooldridge, An Introduction to Multi-Agent Systems. John Wiley & Sons, 2002.

[2] T. Erl, Service-oriented architecture: concepts, technology, and design. Upper Saddle River, NJ: Prentice Hall Professional Technical Reference, 2005.

[3] C. A. Marín et al., "A Conceptual Architecture Based on Intelligent Services for Manufacturing Support Systems," in IEEE International Conference on Systems, Man, and Cybernetics. IEEE Computer Society, oct 2013, pp. 4749–4754.

[4] C. A. Marín, "Evaluation of an Ecosystem-Based Architecture for Adaptation to Cascades of Changes," in IEEE International Conference on Systems, Man, and Cybernetics. IEEE Computer Society, oct 2013, pp. 4760–4765.

[5] C. A. Marín et al., "Application of Intelligent Service Bus in a Ramp-up Production Context," in Proceedings of the Industrial Track of the Conference on Advanced Information Systems Engineering 2013 (CAiSE'13), CAiSE-IT 2013, V. Pelechano, G. Regev, and Y. Pigneur, Eds. CEUR Workshop Proceedings, aug 2013, pp. 33–40.

[6] J. Mendes, P. Leitão, F. Restivo, and A. Colombo, "Service-Oriented Agents for Collaborative Industrial Automation and Production Systems," in Holonic and Multi-Agent Systems for Manufacturing, ser. Lecture Notes in Computer Science, V. Mařík, T. Strasser, and A. Zoitl, Eds. Springer Berlin Heidelberg, 2009, vol. 5696, pp. 13–24.

[7] J. Barbosa and P. Leitao, "Enhancing service-oriented holonic multi-agent systems with self-organization," in International Conference on Industrial Engineering and Systems Management, Metz, 2011, pp. 1373–1381.

[8] D. A. Chappell, Enterprise service bus, 1st ed. Sebastopol, Calif: O'Reilly, 2004.

[9] C. A. Marín, I. Stalker, and N. Mehandjiev, "Engineering Business Ecosystems using Environment-Mediated Interactions," in Engineering Environment-Mediated Multi-Agent Systems: Selected, Revised and Invited Papers, ser. Lecture Notes in Artificial Intelligence (LNCS), D. Weyns, S. Brueckner, and Y. Demazeau, Eds. Heidelberg: Springer-Verlag, Jul. 2008, vol. 5049, pp. 240–258.

[10] J. Otsuka, "A Theoretical Characterization of Ecological Systems by Circular Flow of Materials," Ecological Complexity, vol. 1, no. 3, 2004, pp. 237–252.

[11] A. Rossberg, H. Matsuda, T. Amemiya, and K. Itoh, "An explanatory model for food-web structure and evolution," Ecological Complexity, vol. 2, no. 3, Sep. 2005, pp. 312–321.

[12] F. Lécué and N. Mehandjiev, "Seeking Quality of Web Service Composition in a Semantic Dimension," IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 6, jun 2011, pp. 942–959.

[13] G. A. Lopardo, J. de la Rosa i Esteva, and N. Hormazábal, "A Monitor for Digital Business Ecosystems," in Artificial Intelligence Research and Development, ser. Frontiers in Artificial Intelligence and Applications. Amsterdam: IOS Press, 2007, pp. 77—86.

[14] C. A. Marín, G. Lopardo, and N. Mehandjiev, "An Ecosystem-based Analysis of the Impact of an Interoperability Tool to a Network of SMEs," in 20th International Conference on Collaboration Technologies and Infrastructures, WETICE 2011. IEEE Computer Society, jun 2011.

[15] J. Barbosa and P. Leitao, "Simulation of multi-agent manufacturing systems using Agent-Based Modelling platforms," in International Conference on Industrial Informatics. IEEE, jul 2011, pp. 477–482.

[16] J. Rao and X. Su, "A Survey of Automated Web Service Composition Methods," in Semantic Web Services and Web Process Composition, ser. Lecture Notes in Computer Science, J. Cardoso and A. Sheth, Eds. Springer Berlin Heidelberg, 2005, vol. 3387, pp. 43–54.

[17] I. Al-Oqily and A. Karmouch, "A Decentralized Self-Organizing Service Composition for Autonomic Entities," ACM Transactions on Autonomous and Adaptive Systems, vol. 6, no. 1, Feb. 2011, pp. 1–18.

[18] G. Briscoe and P. De Wilde, "Digital Ecosystems: Evolving Service-Orientated Architectures," in BIONETICS'06: Proceedings of the 1st international conference on Bio inspired models of network, information and computing systems. New York: ACM Press, 2006.

[19] U. Wajid, C. A. Marín, and N. Mehandjiev, "Optimizing Service Ecosystems in the Cloud," in The Future Internet, ser. Lecture Notes in Computer Science (LNCS). Heidelberg: Springer-Verlag, may 2013, vol. 7858, pp. 115–126.

[20] C. Villalba, M. Mamei, and F. Zambonelli, "A self-organizing architecture for pervasive ecosystems," in Self-Organizing Architectures, ser. Lecture Notes in Computer Science. Heidelberg: Springer-Verlag, 2010, vol. 6090, pp. 275–300.

 ISBN: 978-1-61208-391-9