Interpreting and Modeling Baggage Handling System as a System of Systems

Abbas Khosravi, IEEE member, Saeid Nahavandi, IEEE Senior Member, and Doug Creighton
Centre for Intelligent Systems Research, Deakin University, Vic, 3217, Australia
{akhos, nahavand, dougc}@deakin.edu.au

Abstract- The topic of Systems of Systems has been one of the most challenging areas in science and engineering due to its multidisciplinary scope and inherent complexity. Despite all attempts carried out so far in both academia and industry, real world applications are far remote. The purpose of this paper is to modify and adopt a recently developed modeling paradigm for System of Systems and then employ it to model a generic Baggage Handling System of an airport complex. In a top-down design approach, we start modeling process by definition of some modeling goals that guide us in selection of some High Level Attributes. Then Functional Attributes are defined which act as ties between High Level Attributes (the first level of abstraction) and low level metrics/measurements. Since the most challenging issues in developing models for System of Systems are identification and representation of dependencies amongst constituent entities, a machine learning technique is adopted for addressing these issues.

I. INTRODUCTION

In recent years the topic of System of Systems (SoS) has been recognized as one of the most challenging research directions in both academia and industry. Although so far practitioners from neither academia nor industry have reached a universal consensus on definition of SoS [1], there are many cases that fall into SoS frameworks in both military and civil sectors. Well known examples are stock market, transportation system, future combat systems, big manufacturing enterprises, and autonomous robots [2]. While there are real demands for applications of SoS in both military and civilian sectors, its applications still are rare due to lack of well developed theories and techniques for developing and engineering SoS. Beholding the real need for having a standard framework for developing SoS models, a great deal of effort has been made to determine what exactly an SoS means, what its traits are, what it covers, and what it can do for us. Independence in operation, geographical distribution, emergent and evolutionary behaviors are of the most important characteristics of an SoS that have been counted for in literature. Features and traits of an SoS have been explained in [3] and latter restated and discussed with slight modifications in [4]. Within an SOS, constituent systems are geographically distributed and can operate independently. Behaviors of highly interconnected homogenous and heterogenous entities are time varying and their collaboration may result in emergent behavior. Majority of man-made systems have traits of an SoS which makes their management and control a hard task.

Some researchers have tried to develop explicit models for each constituent entity of an SoS and then link these models to create a form of complete picture of the underlying SoS [5]. Although they well model interactions amongst constituent entities, the whole design in this work and similar research is based on this assumption that interactions are not only a priori determined for the modeler, but also remain fixed during SoS operation. Of course this assumption does not scale up to those complex systems composed of several homogeneous or heterogeneous elements with unknown, time variant interactions.

Apart from the difficulties that we confront to engineer systems with different natures that may operate on different time scales, there are also some other vexing issues that come to life when we run simulation models together. Even if we assume that our current level of science is sufficient for engineering each constituent element of an SoS individually, still we are missing some techniques for linking and running all of these separately engineered models as a whole. As indicated in [6], the poor performance of SoS applications and complex systems has more roots in insufficient attention to interactions of systems rather than inadequate engineering of individual systems.

Generally there are a couple of views toward modeling system of systems, network of networks, or complex systems (hereafter we use all these terminologies interchangeably as some researchers define and consider them the same): hierarchical mappings, uncertain state equations, nonlinear mechanism, and autonomous agents [7]. Despite all the attempts made so far for adopting these approaches, none of them have been able to completely describe and model all complexities that we confront in the context of a complex system such as big manufacturing enterprises or health care systems. Furthermore, successful implementation of proposed ideas and tentative plans for modeling complex systems could take many years as we need to solve many problems to obtain an overarching architecture [8]. Based on the current level of expertise in science and engineering as well as research trends, we can consider this opinion that developing models for addressing all aspects of complex systems is far remote.

Recently some researchers have tried to model, represent and, predict performance of a team of Unmanned Aerial Vehicles (UAVs) through the concept of SoS [9]. In a top-down goal-driven approach, they investigate and model only those parts of an SoS which pertain to modeling goals, considerably reducing the level of complexity that they
confront in modeling process. Based on defined modeling goals, a couple of high level attributes are selected from a priori determined set that act as ties between high level modeling goals and low level measurements/metrics. In their approach, they make provision against precedence and causality relationships amongst constituent entities and adopt some techniques to extract and learn them from available data. Although the SoS definition that they have used in their research differs from the one we use in this paper (in terms of SoS traits), their modeling paradigm deserves more investigation. The proposed method does not suffer from limitations mentioned in other methods even in definition of SoS boundary \[10\]. For sure, this does not imply that the proposed method can be successfully applied to all systems with many homogeneous or heterogeneous entities. We just adopt and use it here as a candidate for modeling some technically complex systems that we have done research about them in the past years. In this way we hope we can add more value and knowledge to the SoS research domain and come up with some tentative modeling and analysis methodologies. These may pave the way for civilian sector applications of SoS rather than just its military ones which have been the main focus of SoS research at least in the last decade. Finally, we hope this research can disseminate our ideas to experts from other disciplines who are somehow dealing with SoS, even if they have never looked at the SoS modeling process in this fashion. This may at least soon result in developing a standard taxonomy for the SoS community.

The remainder of this brief is organized as follows. In section 2, we briefly argue that BHS has the main traits of an SoS. The paradigm for modeling an SoS is explained in section 3. In section 4, an SoS model for Baggage Handling System (BHS) is developed. Finally, in section 5, we conclude with a discussion and some points for future work.

II. INTERPRETING BHS AS AN SoS

SoS traits as explained in \[4\] are operational and marginal independence of entities, geographical distribution, emergent behavior, evolutionary behavior, networks, heterogeneity, trans-domain. In this section, we argue that BHS has all those characteristics and traits and therefore can be considered as and analyzed like an SoS. As emergent and evolutionary behaviors are the main features of an SoS, the main focus of the following discussion is on them.

A generic BHS (Figure 1) includes many dumb and smart entities \[12\] which operate together to achieve system processing objectives. Despite their skillfully engineered arrangement in the format of a BHS, constituent homogeneous and heterogeneous entities can operate independently based on their own objectives (e.g., minimum queue length). There are known and unknown dependencies amongst constituent entities which are quite nonlinear (e.g., effects of load balancing policies or failure redundancy provision). While some dependencies are physical (conveyors between two stations), sometimes they are just a matter of information exchange (e.g., data collected and shared by sensors mounted along conveyors). Decisions made by smart entities or operations completed by dumb elements can considerably affect what others decide and do. Many of these highly nonlinear dependencies are time variant and problematic to be engineered. As we have a mix of equipment and personals in different sections of a BHS and as there are too many stakeholders related to or involved in managing and running a BHS (e.g., airport managers, airlines as customers, shift managers, area supervisors, machine operators, and maintenance technicians), expertise in a wide variety of disciplines such as engineering and economy are required for optimal and smooth operation of BHS.

Like any other material handling system \[11\], when the number of buffers, conveyors, and processes increase in BHS, emergent behavior could be observed in term of route congestion, conveyor stoppage, or long travel time of bags. Furthermore, humans (and, in general, any other smart entity) taking part in different processes (e.g., picture reviewing, hand searching, and manual encoding) behave differently, even if they have been exposed to similar trainings and carry similar experiences \[12\]. Proliferation of distributed control systems in BHS and, in general in any material handling system or manufacturing enterprise, could also yield to some undesirable situations that can not be easily handled.

Generally, the provision of pushers for load balancing and failure redundancy, especially before system bottlenecks, is essential for smooth operation of any BHS. Dispatching rule for each pusher may change over time upon operational

![Figure 1](image-url)
conditions that managers confront. The dramatic consequences of such a change could range from occurrence or disappearance of blockages in some routes to enhancing/decreasing overall performance of system. Besides, as behavior of operators and the way they run system change over time, there is a gradual yet continuous evolution in system operation.

Taking into account our previous experiences on developing simulation models for BHSs [13][14], we believe that the BHS, as a technically distributed complex system, could be a great platform/testbed for analyzing, studying, and developing some paradigms for modeling SoSs.

III. SoS Modeling Paradigm

In this section we briefly describe the proposed SoS modeling paradigm in [9]. In a top-down modeling approach, the modeling process starts by definition of the underlying SoS, what we are aiming to model and its boundaries. Then modeling goals are determined to limit the scope of modeling process. In fact through definition of modeling goals, we somehow determine those aspects of the underlying SoS that are matter of interest or importance for us. Through consulting with system managers, we choose High Level Attributes (HLAs) that are directly related to the modeling goals. Selection of HLAs is from a priori defined set which includes the following attributes: command (control), operation (decision making procedures), communication (data exchange), active effects (those environmental systems that affect the SoS operation based on its current condition), passive effects (those environmental systems that their conditions has effects on SoS operation), perception (capability of entities in the SoS to gather and interpret data), and memory (capability of recording data). The first three HLAs are called the internodal interactions. The forth and fifth HLAs are extranodal interactions since those belong to the environment surrounding the SoS. The last two ones are called information-related attributes as they pertain to measurement and data sharing.

After selection of useful HLAs for modeling process, some Functional Attributes (FAs) are defined which act as ties between modeling goals and low level metrics/measurements. These quantitative values could be a function of as many as homogeneous or heterogeneous metrics for measuring an aspect of the SoS operation such as flexibility or agility. Again defining and measuring them is totally up to modelers and considered set of HLAs. These functional attributes could be related to internal operation of an SoS (internodal interactions), related to its active and passive interactions with environment (extranodal interactions), or related to information and data sharing in the SoS. Each FA is measured based on metrics used in its definition. Here metric is a low level measurement which is often benchmarked against a maximum/minimum value.

In the last stage of design paradigm, dependencies amongst metrics constituting a FA, between FAs, and between FAs and metrics of other FAs are searched. This could be carried out through a couple of tools such as regression analysis and other powerful data mining approaches. Dynamic Bayesian Network (DBN) is the powerful tool employed in [9] for finding and representing these time variant, highly nonlinear dependencies. Conducting this stage completes the SoS picture as shown in Figure 2. All dependencies shown in this figure are found using DBN in a blind manner. Modeler starts with a chosen network of dependencies amongst nodes and then adds or removes links based on available data. DBN is able to find and model dependencies amongst all entities of the underlying SoS through updating its structure (links between different nodes) and conditional probabilities [15].

The focus of the proposed paradigm in [9] is on identification of interactions amongst constituent entities of an SoS. In general, paradigm tries to adopt some machine learning approaches for finding and representing relationships. Identified relationships can be simply updated and changed as more data becomes available or new situations occur in the underlying system. The spirit of proposed approach complies with the arguments made in [6] that enough care must be taken for discovering and interpreting dependencies when developing a model for an SoS.

![Fig. 2. SoS representation using the concept of HLAs, FAs, and metrics. In this SoS model, communication and passive effects HLAs have not been included in the final SoS model. While the dotted lines represent interactions amongst different FAs from the three subspaces, the dashed ones show dependencies amongst metrics in three subspaces.](image_url)
IV. Modeling BHS as an SoS

In the two previous sections, we interpreted BHS as a technically complex system that has main characteristics of an SoS and then briefly explained a recently developed SoS modeling paradigm. In this section, the introduced paradigm is employed for modeling BHS as an SoS. To get insights into the operation of BHS, some previously developed and validated stochastic, discrete event simulation models have been used [13][14].

A. SoS definition

In our design, we consider BHS as an SoS with all machines, operators, and processes as constituent entities. All smart and dumb entities that are involved in different processes such as checking-in, tag reading (automatic or manual), hand searching, screening, and bag collecting (lateral and makeup loops) are considered in the SoS model. External entities and the environment surrounding BHS include airport managers, airlines, government authorities, and so on. Carrying a wide variety of objectives such as financial benefits, security, and satisfactions, these systems actively and passively affect operation of BHS.

B. Modeling goal

The purpose of the model is to evaluate performance of a generic BHS. We will consider all those aspects of BHS and other external parties which are related to the modeling goals. Some important performance indexes for BHS managers and decision makers are throughput of the system in each time unit, time required for processing a specific percentile of each flight bags, and utilization of BHS bottlenecks in peak hours. For the sake of simplicity, we only here choose tracking the throughput of the system per time unit which it reflects smooth/problematic operation of the underlying BHS. In future and as research proceeds in our group, we will consider more performance indexes in the modeling goal set.

C. Selection of HLAs based on SoS definition

As explained in the previous section, the set of HLAs includes seven attributes that we only choose those ones which are related to modeling goals and behaviors/patterns that we observe in BHS. Discussion on their selection is given below:

Command (CMD): Many entities of a BHS are controlled and commanded by some distributed controllers which are again linked to a central controller. While the majority of conveyors and mergers are controlled locally based on some feedbacks from the status of the following conveyors or machines, some others such as pushers are controlled centrally for better load balancing. Besides, bag route planning is a key controlling issue performed by central controller as soon as the tag reading process is complete. Therefore, we keep this attribute in the metric space.

Operations (OPS): Majority of smart entities in BHS make decisions based on their own objectives, system configuration, and feedbacks that they get from current operation condition of the system. X-ray machines in the first and third levels of security screen bags and detect explosives or other potential dangers. Bags failing the first or third level of security are visually inspected by some trained employees. As completion of these tasks has a considerable influence on system performance, we keep this attribute in model as well.

Communication (COM): As majority of entities in the underlying SoS have some limited communications to each other and often get information that they need from a central database, this attribute could be discarded in the current model of BHS.

Active Effects (ACT): Almost all those stakeholders that can impact on BHS operations actively are a part of BHS and could be considered within its boundaries. Within this context, we drop out this attribute from the metric space.

Passive Effects (PAS): In contrast, there are many external parties that seek their own objectives, yet considerably affect operation of BHS. Decisions made by airport mangers to gain more profit from airport complex, change in government policies, changes in aircraft types, and applying new bag checking in policies all indirectly affect BHS operation. Due to the passive nature of these effects, there is no guarantee that the consequences will be always desirable.

Perception: There are many entities in the system that gather and interpret data and information. Queues’ lengths, screening machines’ number and their capacities, and number of operators or picture reviewers in each shift work are examples of data shared among constituent entities and stakeholders. As perception and interpretation of the collected data affect system operation and greatly contribute to its satisfactory or poor performance, this attribute is kept in the model.

Memory: Since recording and using data for reasoning and modifying controlling rules is not a much common practice in today’s BHSs, this attribute is left out from the metric space and is not a part of our model.

D. Selection of HLAs based on modeling goal

In this section, we again review attributes selected in the previous stage and remove some of them based on interpretation of modeling goals. Only those attributes are kept in the model that affect the chosen performance index (throughput of the system per time unit). Those not much contributing to the variation of performance measure are considered redundant and removed from the metric space.

Command (CMD): Based on explanations provided before, we may conclude that there is a clear dependency between this attribute and system modeling goals. So we keep this attribute in metric space.

Operations (OPS): Since poor or excellent system performance is attributable to the decisions made by constituent entities of the underlying BHS, we may leave this attribute in as well.

Passive Effects (PAS): This attribute is kept in the model due to severe impacts of decisions made by external parties on the BHS status.
Since data collection, process and distribution are secure and almost noise-free, this attribute has approximately no effect on the performance of this system. Nevertheless to say, if the modeling goals are changed or a more detailed model is requested, more attributes can be defined and kept in the metric space.

### 5. Functional Attributes (FAs)

So far we have kept the following HLAs in the SoS model: CMD, OPS, and PAS (three out of seven). Now we define a couple of useful FAs for each HLA again directed by SoS definition and considered modeling goal. As indicated in [9], useful metrics for measuring a FA directly pop up from its definition. Although researchers in [9] have used some previously well defined and documented functional attributes [17], definition and measurement of functional attributes is still far from being an exact science and requires cross-fertilization between many disciplines such as engineering, economy, and management [18]. Table 1 includes a couple of FAs considered in our model and their metrics. To the best of our knowledge, this is the first attempt to define and measure some attributes for BHS that reflects its operational conditions. Metrics mentioned in the third column of the table are almost illustrative than comprehensive. As research goes ahead, more will be considered for better and more accurate calculation of FAs. To take into account importance of some metrics, the weighted mean of metrics could be used for calculation of each FA. The more important a metric is, the bigger its weight will be in the mathematical formula of the underlying FA.

### F. Dependency Detection and Modeling

The last stage in the modeling process is training a DBN to find dependencies amongst FAs and metrics. Since dispatching rules and many operational parameters such as number of picture reviewers or screening capacity of system may change over time, the trained DBN will have variant

---

### Table I

<table>
<thead>
<tr>
<th>HLA</th>
<th>FA</th>
<th>Useful Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tracking &amp; Routing</td>
<td>Rate of lost bags in the system / Rate of checked in bags, Rate of misrouted bags in the system / Rate of checked in bags, Rate of delayed bags in the system / Rate of checked in bags</td>
<td></td>
</tr>
<tr>
<td>Optimization of Routing</td>
<td>Average travel time / Maximum travel time recorded within a time period, Frequency of rout changing occurrence for a bag, Active rout selection criteria (minimum length, travel time, etc)</td>
<td></td>
</tr>
<tr>
<td>Failure Rate</td>
<td>Failure rate of first (second) X-ray machines / Maximum acceptable rate, Failure rate of tag reading machines / Maximum acceptable rate, Failure rate of important resources (buffers before X-ray machines) / Maximum acceptable rate</td>
<td></td>
</tr>
<tr>
<td>Security Threat</td>
<td>Rate of rechecks by first picture reviewers, second X-ray machines, second picture reviewers, and hand searching stations all divided by rate of checked-in baggage</td>
<td></td>
</tr>
<tr>
<td>Flexibility</td>
<td>No. of possible routes for directing a bag / Maximum no of routes between entry and exit points Possibility of changing priorities in each merging points based on WIP</td>
<td></td>
</tr>
<tr>
<td>Utilization</td>
<td>No. of bags processed by first (second) X-ray machines / Screening capacity of first (second) X-ray machines No. of pictures reviewed by first (second) picture reviewers / Reviewing capacity of first (second) picture reviewers No. of bags searched manually by operators / Hand searching capacity of operators with the highest degree of safety No. of bags encoded manually by operators / Manual bag encoding capacity No. of bags collected by each operator in laterals or make up loops / Collecting capacity of each operator No. of check-in pier in operation / Total no of check-in piers</td>
<td></td>
</tr>
<tr>
<td>Blockage Threat</td>
<td>Check in rate of bags / Total screening capacity of the system Utilization of buffers before X-ray machines / Maximum length of queue Queue length for first (second) X-ray machines, Queue length for first (second) picture reviewers, Queue length for hand searching stations, Queue length in manual encoding station all normalized by maximum allowed length of queue Throughput rate / Rate of checked-in baggage Off-/Peak period (y/n)</td>
<td></td>
</tr>
<tr>
<td>Processing Difficulty</td>
<td>Rate of bags with abnormal size requiring manual handling / Rate of checked in bags Rate of overweight bags / Rate of checked in bags</td>
<td></td>
</tr>
<tr>
<td>Security Level</td>
<td>Current level of security / Highest level of security</td>
<td></td>
</tr>
<tr>
<td>Operator Availability</td>
<td>Current no. of staff in operation / Standard no. of staff required for normal operation of BHS No. of staff added to system in peak hours / Standard no. of staff required for normal operation of BHS Current no. of first (second) picture reviewers / Maximum no. of first (second) picture reviewers Current no. of operators for manual encoding / Maximum no. of operators for manual encoding Current no. of operators in operation / Standard no. of operators required for normal operation of BHS</td>
<td></td>
</tr>
<tr>
<td>Airport Management</td>
<td>No. of delayed bags due changing no. of operators (or other parameters) / No. of delayed bags in normal condition Current no of staff in operation / standard no. of staff required for normal operation of BHS No. of staff added to system in peak hours / standard no. of staff required for normal operation of BHS Time for opening check in counters and Time for closure of laterals and make up loops</td>
<td></td>
</tr>
<tr>
<td>Security Policies</td>
<td>Increment percentage in the rates of rechecks in higher levels of security in comparison to the normal rates / Maximum Increment percentage, Increasing time of hand searching / Maximum Time of hand searching, No. of bags which are considered dangerous after applying new security policies / All bags Increment in no. of staff required for applying new policies / Total no. of staff</td>
<td></td>
</tr>
<tr>
<td>Airline</td>
<td>Flight type (economy or economy-business), Seat free flights</td>
<td></td>
</tr>
</tbody>
</table>
structure and time varying conditional probabilities. Starting by a simple network, the learning method will update network structure and appropriately weigh all dependencies.

Due to lack of training data for some of the metrics mentioned in Table 1, we will accomplish this part in the near future and report the obtained results in other publications. The final SoS model of BHS will be similar to the one represented in Figure 1 with three HLAs (CMD, OPS, and PAS). As we change the modeling goals and consider other aspects of BHS, we will require additional metrics and therefore the structure of the trained DBN will change. Once the DBN is trained offline, it can be used online for predicting any metric or FA of interest. Network structure and conditional probabilities will graphically and mathematically represent their dependencies and evolution over time.

In the practice of training a DBN, learning all combinations of possible situations to cover all the regions of input space is to some extent impossible. So, the concept of Design of Experiment (DOE) becomes important here and will be used for data collection. Furthermore, to guarantee reasonable operation of the developed model outside its training regions, some self-modeling techniques will be adopted for creating resilient machines, a domain with many critical issues left unarticulated.

The utilized method in this research can also be applied for modeling the whole airport complex through considering more HLAs and defining useful FAs. Another interesting domain for implementation of this kind of SoS model development is space exploration, in particular for those cases that understanding and modeling relations between autonomous robots and human operators involved in a mission is not straightforward.

The goal-driven top-down design approach employed here could also be used in other domains, especially for those cases that due to inherent complexity of the underlying SoS, finding and representing dependencies amongst constituent systems is quite problematic. It may be wisely tailored for use in specific application areas such as modeling human-robot interactions, supply chain models, and manufacturing enterprises.

REFERENCES