

10-23-2017

Sheaf Theory Approach to Distributed Applications: Analysing Heterogeneous Data in Air Traffic Monitoring

Seyed M.H. Mansourbeigi
Utah State University, phy_math_ee@yahoo.com

Follow this and additional works at: https://digitalcommons.usu.edu/computer_science_stures

 Part of the [Computer Sciences Commons](#)

Recommended Citation

Seyed Mansourbeigi, Sheaf Theory Approach to Distributed Applications: Analysing Heterogeneous Data in Air Traffic Monitoring, *International Journal of Data Science and Analysis*. Vol. 3, No. 5, 2017, pp. 34-39. doi: 10.11648/j.ijdsa.20170305.11

This Article is brought to you for free and open access by the Computer Science Student Works at DigitalCommons@USU. It has been accepted for inclusion in Computer Science Student Research by an authorized administrator of DigitalCommons@USU. For more information, please contact rebecca.nelson@usu.edu.



Sheaf Theory Approach to Distributed Applications: Analysing Heterogeneous Data in Air Traffic Monitoring

Seyed Mansourbeigi

Department of Computer Science, College of Engineering, Utah State University, Logan, USA

Email address:

smansourbeigi@aggiemail.usu.edu, phy_math_ee@yahoo.com

To cite this article:

Seyed Mansourbeigi. Sheaf Theory Approach to Distributed Applications: Analysing Heterogeneous Data in Air Traffic Monitoring. *International Journal of Data Science and Analysis*. Vol. 3, No. 5, 2017, pp. 34-39. doi: 10.11648/j.ijdsa.20170305.11

Received: September 6, 2017; **Accepted:** September 23, 2017; **Published:** October 23, 2017

Abstract: The goal of the present article is to demonstrate a mathematical modeling for distributed applications. The present paper applies tools from topology and sheaf theory as an appropriate mathematical modeling to reflect interactions among elements of distributed applications resources. Sensors are characterized from their topological representations in distributed network system. This modeling is applied for the study of the air traffic monitoring system and discuss the model in detail.

Keywords: Cellular Sheaf, Stalks, Cosheaf Homology, Sheaf Cohomology

1. Introduction

The biggest engineering problems are fun math problems. So if you have hard engineering problem and you cannot crack, it almost always has a neat mathematics problem buried under there.

Distributed applications are applications or software that runs on multiple computers within a network at the same time and can be stored on servers. Data management is a key aspect of any distributed system. One of the main challenges in today's computer science research is the extraction of information from heterogeneous datasets. There have been numerous research that have shown the impact of mathematics in network modeling. The challenge is to understand how data is organized by turning data into information; information into knowledge; and eventually knowledge into wisdom. Geometry and topology are the natural modern approaches to handle complex heterogeneous data. The computations presented in this paper yields towards a bridge between modern geometry, topology and distributed systems and aims to introduce methods based on geometry and topology to detect and manage particular structures of the complex system. In recent years there has been researches on the application of sheaf theory to provide a semantic foundation for distributed applications [1] [2]. A sheaf can be thought of as a system of observations on a topological space, with the key property that consistent local observations can be uniquely pasted together to provide a global observation.

Application of sheaf theory in computer science has a long historical track. An early use of sheaf theory was a paper by Monteiro and Pereira [3]. They applied sheaf theory to study connections between event systems. As a foundation for the behavior of concurrent processes Ehrlich, Goguen and Sernadas [4] and Goguen [5] and Cattani, G. L. and G. Winskel [6] applied sheaf theory. Cirstea [7] provided the semantics for a concurrent object-oriented programming language using sheaf theory. The motivation in this paper is inspired by the very recent applications of sheaf theory in computer science and software engineering. These applications can be found in [1] [2] [8] [9].

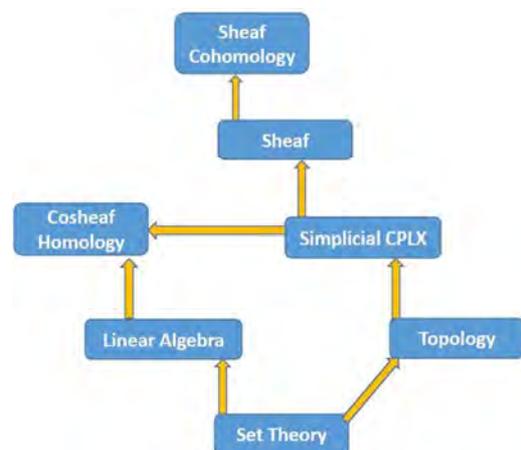


Figure 1. Road Map towards the Creation of the Modeling.

The paper will address problems like management of heterogeneous data, behavior of data, failure detection, and noise in data exchange. In this regard techniques borrowed from topology and geometry are explored to treat distributed applications. Figure 1 shows the road map towards the creation of the modeling.

The scheme of the present paper rests on the following tasks:

1. Topologization: Develop topological modeling for distributed data systems to create the basin space for data.

2. Analysis of Data (Encoding existing data into sheaves: Utilizing sheaf theory method to enable exchange of the heterogeneous data and to evaluate the behavior of the system.

3. Example: The developed modeling and computations is applied to analyze the air traffic monitoring distributed system.

2. Topology and Sheaf Modeling

2.1. Swimming in Sensors, Drowning in Data

This section begins with definitions and constructions of the topological modeling of the distributed applications. Topology approach modeling is applied to reflect interactions among data sources in a network. Sheaf theoretic modeling approach is to demonstrate relations between these data sources. Both approaches are towards better understanding of the distributed applications.

2.2. Simplicial Complexes

Main pillar of the whole construction is the notion of data space, the crucial feature of which is a topological space. Fundamental goal of it is to simply converting the collection of points in data space into a combinatorial format; a simplicial complex model. Simplicial complex is an object whose edges are determined by some given notion of communication between nodes (sensors). Simplicial complexes are very useful generalizations of graphs and can be used to model interactions involving more than just two nodes in a distributed application.

Definition 1.

A simplicial complex \mathcal{K} consists of points (vertices), line segments (edges), triangles and their n -dimensional generalizations (n -simplices: polytopes as the convex hull of $n+1$ vertices). The convex hull of any nonempty subset of the $n+1$ points of an n -simplex is called a face of the simplex.

More precisely, a simplicial complex \mathcal{K} is a set of simplices satisfying the following conditions:

1. Any face of a simplex belonging to \mathcal{K} is also in \mathcal{K} .
2. The intersection of any two simplices in \mathcal{K} is either the empty set or is another simplex in \mathcal{K} .

Figure 2 shows an example of a simplicial complex.

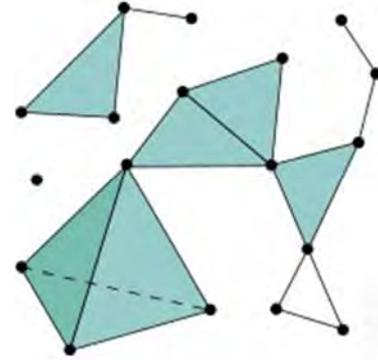


Figure 2. An Example of a Simplicial Complex \mathcal{K} .

The concept of simplicial complex is applied in a distributed network system modeling by the following assignments:

1. The vertices of the simplicial complex represent the nodes (attributes like hosts, network devices, sensors, etc.) in a distributed application.

2. The communication and exchange of data between the nodes (Cables, RG45, Microwave Dishes, WIFI, and Satellite Communications) in a distributed application are represented by n -simplices ($n \geq 1$) in a simplicial complex.

The deficiency with this model is that the simplicial complex approach tends to rely on the homogeneity of information sources. For a consistency between observations and encoding the interactions among heterogeneous information sources application of a stronger tool is needed. Details about integration of heterogeneous data and encoding their interactions is explained as follows.

2.3. Cellular Sheaves

Various problems in distributed systems such as rerouting information flow around a failed subsystem can have interpretations in a more generalized mathematical theories [10]. The cell complex approach relies on the homogeneous data and for integrating heterogeneous data a more powerful application is required to analyze dissimilar data types. The application of sheaf theory provides an appropriate approach involving a heterogeneous set of data sources. So the next step would be encoding the existing heterogeneous data from the simplicial complex into a sheaf model. To each simplex in the complex there assigns a set which carries all information about the data and enables us to analyze data transmission in distributed network system [11].

Definition 2.

Let \mathcal{K} be a simplicial complex. A cellular sheaf F over \mathcal{K} is an assignment of an information space (vector space) $F(\sigma)$ to each simplex σ in \mathcal{K} called the stalk of the simplex σ such that:

1. For any two simplices σ and τ in \mathcal{K} with σ as the immediate face of τ ($\sigma < \tau$), there exists a restriction map $F(\sigma \rightarrow \tau): F(\sigma) \rightarrow F(\tau)$.
2. For every three simplices with $\sigma < \tau < \omega$; $F(\sigma \rightarrow \tau) \circ F(\tau \rightarrow \omega) = F(\sigma \rightarrow \omega)$.

Figure 3 shows an example of a simplicial complex

together with the assignments of stalks and the restriction maps.

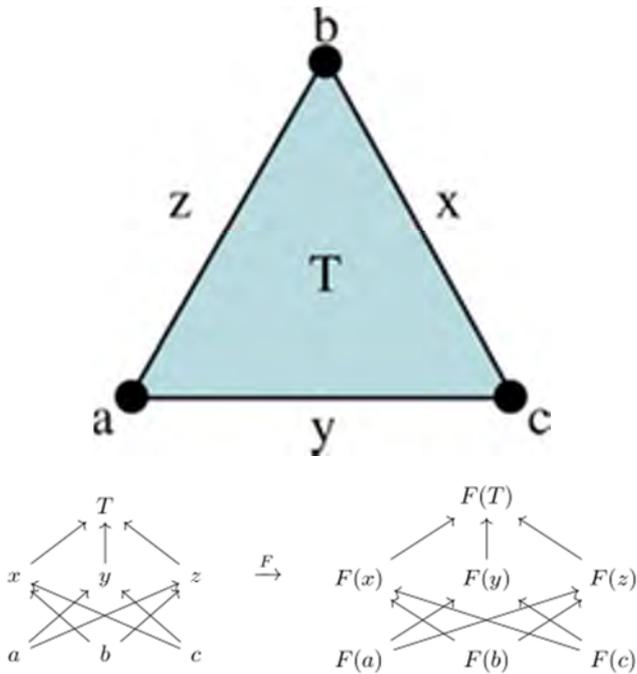


Figure 3. An Example of a Simplicial Complex \mathcal{K} Together with the Associated Sheaf F . Inclusions of the faces are Shown by Upward Arrows.

3. Example: Air Traffic Monitoring

Air traffic monitoring is one of the crucial and critical complex system to detect and estimate the location, velocity and flight direction of a large number of various airplanes approaching the airport. In air traffic monitoring in an airport, multiple sensors of various types are monitoring the region. To make the detection more precise consider duplication of the sensors of same type. Consider cluster of GPS satellites, cluster of Radar stations, cluster of Airport Surface Detectors and cluster of smart IR (Infrared) sensors for air traffic monitoring. Figures 4, 5 show examples of air traffic monitoring system. As it is seen, there are a numerous heterogeneous data acquisition needed to be integrated.



Figure 4. An Example of an Air Traffic Monitoring System in Airport.

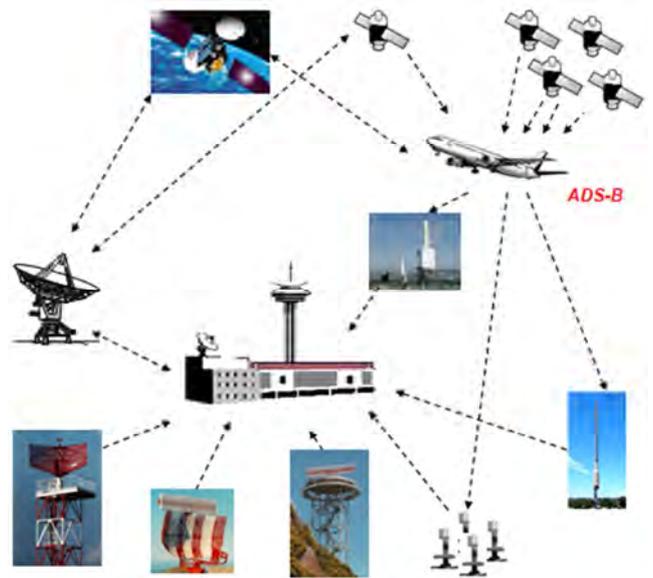


Figure 5. An Example of an Air Traffic Monitoring System.

Table 1. Shows the Types of the Sensors and Their Duplication Numbers.

Table 1. Sensors and Their Duplication Numbers.

Sensor type	Number of Sensors time $t=t_0$
Radars (R)	n
GPS (G)	m
Airport Surface Detectors(K)	p
IR Sensors (I)	q

3.1. Topology and Sheaf Construction

Sensors of the same type communicate and report a common data. The heterogeneous data received at time $t=t_0$ are given in the table 2. The measured subjects in the table are:

1. Aircraft Status (E),
Space of measurement = \mathbb{R}
2. Aircraft Coordinates (C),
Space of measurement = \mathbb{R}^3
3. Direction (D)
Space of measurement = \mathbb{R}^3
4. Speed (S)
Space of measurements = \mathbb{R}

Table 2. The Heterogeneous Data Received by the Sensors.

Sensors vs. Data	(E)	(C)	(D)	(S)
(R)		√	√	√
(G)		√		
(K)	√		√	
(I)	√			√

First step: The construction of the simplicial complex (CPLX) \mathcal{K}

The simplicial complex structure model is shown in Figure 6.

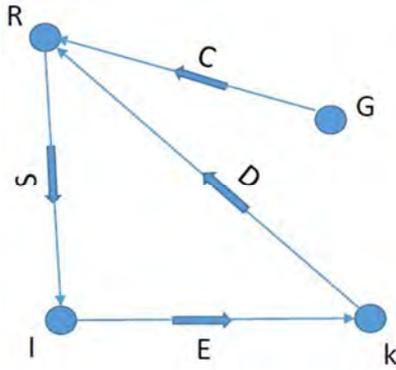


Figure 6. Simplicial Complex Model with Oriented Simplices.

Second step: The sheaf construction

Each simplex in the simplicial complex has a unique ID. This is represented by assignment of additional information to it. To model this assignment, for each simplex in the complex an additional information namely a stalk is assigned. It carries all information about the data and its neighboring nodes and enables us to analyze the system. The assigned stalk's spaces is as follows:

Stalk $R = F(R) = \{\text{Aircraft Coordinates } \mathbb{R}^3, \text{Direction } \mathbb{R}^3, \text{Speed } \mathbb{R}\} \cong \mathbb{R}^3 \oplus \mathbb{R}^3 \oplus \mathbb{R}$

Stalk $G = F(G) = \{\text{Aircraft Coordinates } \mathbb{R}^3\} \cong \mathbb{R}^3$

Stalk $K = F(K) = \{\text{Aircraft Status } \mathbb{R}, \text{Direction } \mathbb{R}^3\} \cong \mathbb{R} \oplus \mathbb{R}^3$

Stalk $I = F(I) = \{\text{Aircraft Status } \mathbb{R}, \text{Speed } \mathbb{R}\} \cong \mathbb{R} \oplus \mathbb{R}$

Stalk $RG = F(RG) = \{\text{Aircraft Coordinates } \mathbb{R}^3\} \cong \mathbb{R}^3$

Stalk $RK = F(RK) = \{\text{Direction } \mathbb{R}^3\} \cong \mathbb{R}^3$

Stalk $RI = F(RI) = \{\text{Speed } \mathbb{R}\} \cong \mathbb{R}$

Stalk $IK = F(IK) = \{\text{Aircraft Status } \mathbb{R}\} \cong \mathbb{R}$

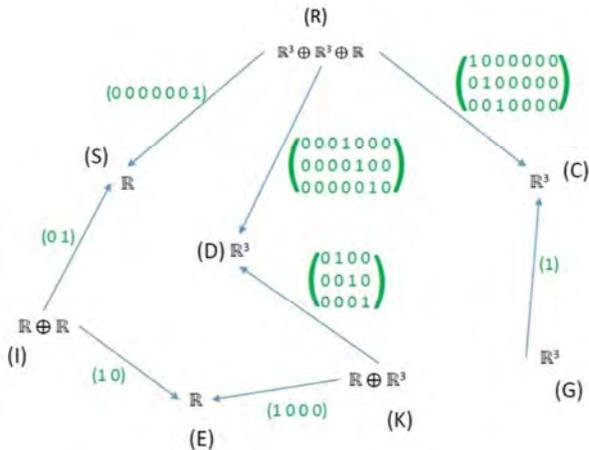


Figure 7. Sheaf, Stalks and Restriction Maps.

Third Step: Computations of Cosheaf Homology and Sheaf Cohomology
For the chain complex

$$C_2 \xrightarrow{d_1} C_1 \xrightarrow{d_0} C_0 \quad (1)$$

With C_k = the vector space generated by the k -simplices, the computation of Cosheaf Homology is as follows:

$$H_k = \frac{\text{Ker } d_k}{\text{Im } d_{k+1}} ; d_k(b) = \sum_{a < b} [b: a] a ; b \in C_k \quad (2)$$

Where for the $(n+1)$ -simplex b and the n -simplex a the incidence number $[b, a]$ is defined as:

$$[b: a] = \begin{cases} 0 & \text{if } a \text{ is not a face of } b \\ (-1)^n & \text{if you delete the } n^{\text{th}} \text{ vertex of } b \text{ to get } a \end{cases} \quad (3)$$

For the cochain complex

$$C^0 \xrightarrow{d^0} C^1 \xrightarrow{d^1} C^2 \quad (4)$$

Where $C^k = \bigoplus \text{Stalk } \sigma = \bigoplus F(\sigma)$, and the sum is over all k -simplices σ , the computation of Sheaf Cohomology is as follows:

$$H^k = \frac{\text{Ker } d^k}{\text{Im } d^{k-1}} ; d^k = \begin{pmatrix} \vdots \\ \vdots \\ \dots [b:a] F(b \rightarrow a) \dots \\ \vdots \\ \vdots \end{pmatrix} \quad (5)$$

Results from calculations

Based on Sheaf, Stalks, Restriction maps at time $t=t_0$, from the algorithm for calculation of the Cosheaf Homology for the simplicial CPLX in Figure 6.

$H_0 = \mathbb{R}$ (dimension of $H_0 = 1$), meaning one connected simplicial CPLX exists.

$H_1 = \mathbb{R}$ (dimension $H_1 = 1$), meaning one hole in this simplicial CPLX in 2D exists.

$H_n = 0$ for $n > 2$, it means in this simplicial CPLX there exists no voids in dimension bigger than 2D.

From the algorithm for calculation of Sheaf Cohomology (Data Analysis):

Dimension $H^0 = 8$, meaning that at time t_0 significant stalks are on R and I , i.e. the global information (section globalization) is extracted from R and I .

$H^1 = 0$, meaning that the Sheaf is a "flasque" sheaf (i.e. restriction maps are projections) and interpreted as no lack of data.

$H^n = 0$ ($n > 1$), since there exists no n -simplices for $n > 1$.

3.2. Time Variation

Suppose at time $t=t_1$ number of i Airport Surface Detectors are out of mission and table 1 has been changed to the table 3.

Table 3. Sensors and Their Duplication Numbers.

Sensor type	Number of Sensors time $t=t_0$
Radars (R)	n
GPS (G)	m
Airport Surface Detectors(K)	$p - i$
IR Sensors (I)	q

As a result the aircraft Status E is no longer detected by K . Table 4. Shows the change that occurs in table 2.

Table 4. The Heterogeneous Data Received by the Sensors.

Sensors vs. Data	(E)	(C)	(D)	(S)
(R)		√	√	√
(G)		√		
(K)	Out of mission		√	
(I)	√			√

The new simplicial CPLX will be as in Figure 8.

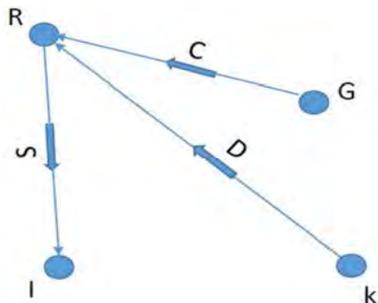


Figure 8. Simplicial Complex Model with Oriented Simplices.

The new stalks would be as:

Stalk R = F(R) = {Aircraft Coordinates \mathbb{R}^3 , Direction \mathbb{R}^3 , Speed \mathbb{R} } $\simeq \mathbb{R}^3 \oplus \mathbb{R}^3 \oplus \mathbb{R}$

Stalk G = F(G) = {Aircraft Coordinates \mathbb{R}^3 } $\simeq \mathbb{R}^3$

Stalk K = F(K) = {Aircraft Status \mathbb{R} , - Direction \mathbb{R}^3 } $\simeq \mathbb{R} \oplus \mathbb{R}^3$

Stalk I = F(I) = {Aircraft Status \mathbb{R} , Speed \mathbb{R} } $\simeq \mathbb{R} \oplus \mathbb{R}$

Stalk RG = F(RG) = {Aircraft Coordinates \mathbb{R}^3 } $\simeq \mathbb{R}^3$

Stalk RK = F(RK) = {Direction \mathbb{R}^3 } $\simeq \mathbb{R}^3$

Stalk RI = F(RI) = {Speed \mathbb{R} } $\simeq \mathbb{R}$

Figure 9. Shows the new sheaf, stalks and restriction maps

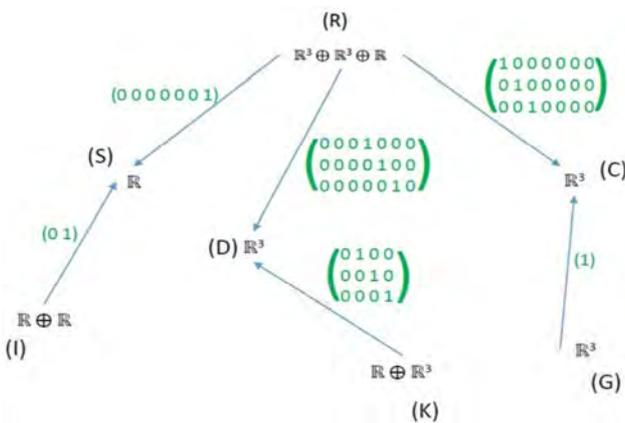


Figure 9. Sheaf, Stalks and Restriction Maps.

Based on Sheaf, Stalks, Restriction maps at time $t=t_1$, from the algorithm for calculation of the Cosheaf Homology for the simplicial CPLX in Figure 8.

$H_0 = \mathbb{R}$ (dimension of $H_0 = 1$), meaning one connected simplicial CPLX exists.

$H_1 = 0$ (dimension $H_1 = 0$), meaning no hole in this simplicial CPLX exists.

$H_n = 0$ for $n > 2$, it means in this simplicial CPLX there exists no voids in dimension bigger than 2D.

From the algorithm for calculation of Sheaf Cohomology

(Data Analysis):

Dimension of $H^0 = 9$, meaning that at time t_1 significant stalks are on K,G and I, i.e. the global information (section globalization) is extracted from K,G and I.

$H^1 = 0$, meaning that t Sheaf is a “flasque” sheaf (i.e. restriction maps are projections) and interpreted as no lack of data.

$H^n = 0$ ($n > 1$), since there is no n-simplices for $n > 1$.

Feedback from the Example

By the change in the number of sensors (sensors become inactive or out of mission), the changes in Cosheaf Homology and Sheaf Cohomology (from time $t=t_0$ to time $t=t_1$) occur. As a result the significant sensors are changed. The dynamic is as follows:

$$H_0(t_0) = \mathbb{R} \rightarrow H_0(t_1) = \mathbb{R} \text{ (Which was expected)}$$

$$H_1(t_0) = \mathbb{R} \rightarrow H_1(t_1) = 0 \text{ (The hole disappears)}$$

$$\text{dimension } H^0(t_0) = 8 \rightarrow \text{dimension } H^0(t_1) = 9$$

(The change in the stalks resulted in the change of significant sensors from {R, I} to {K, G, I}).

Summary and Future Work

The computations presented in this paper yields towards a bridge between modern geometry, topology and distributed applications. The paper focuses on the mathematical modeling for the distributed applications and aims to introduce methods based on geometry and topology to detect and manage particular structures of the complex system. The software for computation of the matrix rank and the image and kernel was “MATLAB”. There are some open problems for future work. Among them are:

1. Addressing large or varied datasets (stalks).
2. Dynamical behavior of the distributed applications.

Acknowledgements

The author would like to express his appreciation for valuable comments from Dr. Stephen W. Clyde (Computer Science Department of Utah State University, USA).

References

[1] M. Robinson, C. Joslyn, E. Hogan, and C. Capraro, “Conglomeration of Heterogeneous Content using Local Topology,” American University, Mar. 2015.

[2] M. Robinson, “Sheaves are the canonical data structure for sensor integration,” Inf. Fusion, vol. 36, pp. 208–224, Jul. 2017.

[3] L. Monteiro and F. Pereira, “A sheaf-theoretic model of concurrency,” in Proceeding of Logic in Computer Science, 1986, pp. 66–76.

- [4] Ehrlich H. D., Goguen J. A., Sernadas A., “A categorical theory of objects as observed processes | SpringerLink.” [Online]. Available: <https://link.springer.com/chapter/10.1007/BFb0019445>. [Accessed: 03-Sep-2017].
- [5] J. A. Goguen, “Sheaf Semantics for Concurrent Interacting Objects,” in *Mathematical Structures in Computer Science*, 1992, pp. 159–191.
- [6] G. L. Cattani and G. Winskel, “Presheaf models for concurrency,” in *Computer Science Logic*, 1996, pp. 58–75.
- [7] Cirstea, C, “A distributed semantics for FOOPS,” University of Oxford, Technical Report PRG-TR-20-95, 1995.
- [8] G. Malcolm, “Component-Based Specification of Distributed Systems,” *Electron. Notes Theor. Comput. Sci.*, vol. 160, pp. 211–224, Aug. 2006.
- [9] G. Malcolm, “Sheaves, Objects, and Distributed Systems,” *Electron. Notes Theor. Comput. Sci.*, vol. 225, pp. 3–19, Jan. 2009.
- [10] Nicholas A. Scoville and Karthik Yegnesh, “Cosheaf Theoretical Constructions in Networks and Persistent Homology,” *International Mathematics Research Notices*, 2009.
- [11] J. M. Curry, “Dualities between cellular sheaves and cosheaves,” *J. Pure Appl. Algebra*, Jun. 2017.